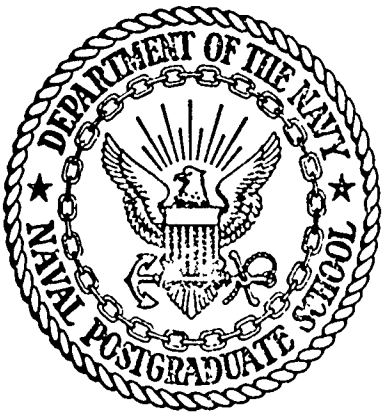
NAVAL POSTGRADUATE SCHOOL
Monterey, California



THESIS

FINITE ELEMENT SOLUTION FOR
AXISYMMETRIC TRANSIENT THERMAL STRESSES

by

Manouchehr Bakhshandehpour

Thesis Advisor: R. E. Newton

June 1972

Approved for public release; distribution unlimited.

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*

Naval Postgraduate School
Monterey, California 93940

2a. REPORT SECURITY CLASSIFICATION

Unclassified

2b. GROUP

3 REPORT TITLE

FINITE ELEMENT SOLUTION FOR AXISYMMETRIC TRANSIENT THERMAL STRESSES

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*

Mechanical Engineer's Thesis; June 1972

5. AUTHOR(S) *(First name, middle initial, last name)*

Manouchehr Bakhshandehpour; Lieutenant, Imperial Iranian Navy

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
June 1972	174	9

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
d.	

10. DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Naval Postgraduate School Monterey, California 93940

13. ABSTRACT

A finite element formulation for solving axisymmetric transient heat conduction and thermal stress problems is developed in this thesis. The governing equations of uncoupled, linear, isotropic thermoelasticity are discretized using quadratic isoparametric elements. A FORTRAN IV program, using double precision arithmetic, is presented. Compact storage techniques for banded symmetric matrices are used.

Comparisons between exact and computer solutions demonstrate close agreement for a number of test problems. Detailed instructions for using the program are included.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6811

UNCLASSIFIED

Security Classification

A-31406

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
AXISYMMETRIC TRANSIENT HEAT CONDUCTION						
FINITE ELEMENT HEAT CONDUCTION						
FINITE ELEMENT TRANSIENT THERMAL STRESSES						
AXISYMMETRIC TRANSIENT THERMAL STRESSES						

Finite Element Solution for Axisymmetric Transient Thermal Stresses

by

Manouchehr Bakhshandehpour
Lieutenant, Imperial Iranian Navy
B.S., Italian Naval Academy, 1960

Submitted in partial fulfillment of the
requirements for the degrees of

MECHANICAL ENGINEER

and

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 1972

Author M. Bakhshandehpour

Approved by: R. E. Newton
Thesis Advisor

Robert K. Mann
Chairman, Department of Mechanical Engineering

Melvin H. Clausen
Academic Dean

ABSTRACT

A finite element formulation for solving axisymmetric transient heat conduction and thermal stress problems is developed in this thesis. The governing equations of uncoupled, linear, isotropic thermoelasticity are discretized using quadratic isoparametric elements. A FORTRAN IV program, using double precision arithmetic, is presented. Compact storage techniques for banded symmetric matrices are used.

Comparisons between exact and computer solutions demonstrate close agreement for a number of test problems. Detailed instructions for using the program are included.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

Note: A single underline is used to denote a column vector and a double underline denotes a rectangular matrix. The symbols used in computer program are described in the beginning of Appendix B.

a	Entrance fluid cross-sectional area
$\underline{\underline{A}}$	Linear combination of $\underline{\underline{C}}$ and $\underline{\underline{Y}}$ matrices
$\underline{b}$	A constant vector
$\underline{\underline{B}}$	Standard rectangular strain-displacement matrix
$\underline{\underline{C}}$	Thermal capacitance matrix
c	Specific heat
$\underline{\underline{D}}$	Standard elasticity matrix
E	Young's modulus of elasticity
e	Superscript designating element contribution
$\underline{F}$	Load vector
$\underline{F}_T^e$	Element thermal load vector
$\underline{F}_P^e$	Element pressure load vector
$\underline{F}_c^e$	Element centrifugal load vector
$\underline{\underline{G}}$	Linear combination of $\underline{\underline{C}}$ and $\underline{\underline{Y}}$ matrices
h	Surface heat transfer coefficient
$\underline{\underline{I}}$	Identity matrix
$\underline{\underline{J}}$	Jacobian coordinate transformation matrix
$\underline{\underline{K}}$	System stiffness matrix
$\underline{\underline{K}}^e$	Element stiffness matrix
k	Thermal conductivity
L	Thickness of slab

ℓ	Arc length along the side of quadrilateral
N_i	Shape function
m	Total number of nodes
n	Outward normal or number of nodes per element
P	Pressure
R	Radial coordinate
S	Surface area
$\underline{T}$	Nodal temperature vector
T	Temperature or, when used as a superscript, transpose of a matrix
T_{avg}	Average temperature
T_f	Fluid temperature (used in one-dimensional example)
$\underline{U}$	A vector defined as $<1\ 1\ 1\ 0>^T$
u	Radial displacement
$\underline{v}$	Right-hand side vector in conduction equation
V	Volume
W	Work done by loads
$\underline{\underline{W}}$	Modal matrix
w	Axial displacement
$\underline{w}$	Eigenvector
$\underline{\underline{Y}}, \underline{\underline{Y}}^+$	Thermal admittance matrix
y	Dependent variable
y^*_{ij}	Element ij of the matrix $\underline{\underline{Y}}^*$
[ ]	Matrix representation
< >	Row vector
$\overline{\nabla}$	Gradient operator
α	Coefficient of thermal expansion
$\underline{\beta}$	Constant coefficient vector

$\underline{\delta}$	Nodal displacement vector
$\underline{\delta}^e$	Element displacement vector
$\underline{\varepsilon}$	Strain vector
μ	Eigenvalue of one-dimensional transient temperature solution
$\underline{\varepsilon}_o$	Thermal strain vector
$\underline{\varepsilon}^e$	Element strain vector
η	Local element coordinate
ν	Poisson's ratio
λ	Eigenvalue
ρ	Material density
$\underline{\sigma}$	Stress vector
ξ	Local element coordinate
τ	Time
$\Delta\tau$	Step size of numerical time integration
τ_{RZ}	Shearing stress component
θ	Fluid temperature, or angle
$\underline{\underline{\Lambda}}$	Spectral matrix
Ω	Speed of rotation

ACKNOWLEDGEMENTS

The author would like to express his gratitude to the Imperial Iranian Navy for having made it possible for him to undertake the course of graduate study leading to the present text.

To Dr. Robert E. Newton, Professor of Mechanical Engineering, the author wishes to express deep appreciation for his inspiring advice and guidance. Without his diligence and patience this study could not have been accomplished in its present form. The author is obliged to Dr. Gilles Cantin, Professor of Mechanical Engineering, for his generous advice during the course of study and computer programming. Thanks are also due to the personnel at the computer center of Naval Postgraduate School.

Finally the author should thank his wife for her patience and understanding throughout his work at this institution.

I. INTRODUCTION

Thermal stresses have become increasingly important in engineering practice during recent years. In power generation higher cycle temperatures and use of nuclear fission are largely responsible for this trend. This thesis describes a computer program for finding temperatures and stresses in bodies having axisymmetric geometry and loading. The governing equations are those of isotropic, uncoupled, quasi-static, linear thermoelasticity. They are discretized by using the finite element method. A FORTRAN IV computer program using double-precision arithmetic has been written to solve problems of the following kinds.

A. TEMPERATURE PROBLEMS

The transient temperature vector, evaluated at the nodal points, may be obtained for an axisymmetric body with the combination of insulated, convection, or constant temperature boundary conditions. For the convection thermal boundary condition, however, we may have fluid flowing with entry temperature prescribed as a linear function of time (RAMP). The program can handle up to 15 different ramps, each having a different flow velocity, and with discontinuities between successive ramps.

B. STRESS PROBLEMS

The program will generate load vectors for pressure loading, centrifugal loading, and axial force. Provision is

made for direct input of one additional load vector. Stresses may be found for any combination of these loadings.

C. THERMAL STRESS PROBLEMS

Thermal stresses may be found for as many as 20 different temperature vectors which may be output of the temperature problem or direct input. In short, in this part any combination of the temperature and stress problems may be used.

II. FINITE ELEMENT FORMULATION OF HEAT CONDUCTION IN AXISYMMETRIC BODIES

A. METHOD OF FORMULATION

For bodies of revolution under axisymmetric loading the mathematical problems presented are two-dimensional. The governing equation for non-steady heat conduction is

$$\bar{\nabla}\cdot k\bar{\nabla}T = \rho c\dot{T}, \tag{1}$$

where k is the thermal conductivity, ρ the density, c the specific heat, T the temperature and $\bar{\nabla}$ the gradient operator. The superior dot denotes a time derivative.

Applying Galerkin's principle [1] gives

$$\int_V N_i \bar{\nabla}\cdot k\bar{\nabla}T \, dV = \int_V \rho c \, N_i \, \dot{T} \, dV, \tag{2}$$

where the integral is over the volume V of the conducting body and N_i is a "shape function" used in representing the temperature distribution. If S is the surface of the body and n the outward normal to surface, then using Gauss' theorem we can write

$$\int_V \bar{\nabla}\cdot(N_i \, k\bar{\nabla}T) \, dV = \int_S N_i \, k\frac{\partial T}{\partial n} \, dS. \tag{3}$$

Since

$$\int_V \bar{\nabla}\cdot(N_i \, k\bar{\nabla}T) \, dV = \int_V (\bar{\nabla}N_i)\cdot(k\bar{\nabla}T)\,dv + \int_V N_i\bar{\nabla}\cdot(k\bar{\nabla}T)\,dV, \tag{4}$$

we can combine Eqs. 2, 3 and 4 to get

$$\int_V \rho c N_i \dot{T}\, dV + \int_V (\bar{\nabla} N_i) \cdot (k \bar{\nabla} T)\, dV = \int_S N_i k \frac{\partial T}{\partial n}\, dS. \qquad (5)$$

Each node of the solid region has a separate discretized linear equation calculated from Eq. 5 using the appropriate shape function N_i. Thus each of the volume integrals on the left hand side of Eq. 5 yields a square coefficient matrix in the assembled set of equations. Calculation of these matrices is a standard process. Details are given by Zienkiewicz [1].

The discretized set of equations takes the form

$$\underline{\underline{C}}\, \underline{\dot{T}} + \underline{\underline{Y}}\, \underline{T} = \underline{v}\ , \qquad (6)^{\checkmark}$$

where there is a term by term correspondence with Eq. 5. The real symmetric matrices $\underline{\underline{C}}$ and $\underline{\underline{Y}}$ represent, respectively, the thermal capacitance and thermal admittance. The elements of the vector $\underline{T}$ are nodal temperatures. The vector $\underline{v}$, discussed in the following section, depends upon the thermal boundary conditions.

In the present development piecewise constant material properties have been assumed, i.e., k, ρ and c are constant within each element, but may vary from element to element. Also two-dimensional isoparametric elements are used. Applicable equations are summarized in Appendix A.

$^{\checkmark}$In this text a double underline denotes a rectangular matrix and single underline denotes a column vector.

B. THERMAL BOUNDARY CONDITIONS

Thermal boundary conditions affect only those scalar equations of Eq. 6 which correspond to boundary nodes. Accordingly, the vector $\underline{v}$ is sparse. Also, in a single problem it is common to have different thermal boundary conditions on individual portions of the boundary. In what follows the subvectors of $\underline{v}$ (distinguished by individual superscripts) which correspond to separate boundary conditions are treated individually.

1. Insulated

It is clear that for insulated boundary conditions the subvector $\underline{v}^{(1)}$ of the right hand side of Eq. 6 corresponding to this boundary condition is zero, since $\frac{\partial T}{\partial n} = 0$.

2. Convection

The heat transfer mechanism occurs in the interface of the solid and fluid. If the fluid temperature is θ and the heat transfer coefficient is h, then equating heat conducted away from the surface to the efflux from the solid [2] gives

$$-k \left(\frac{\partial T}{\partial n}\right)_S = h(T-\theta), \tag{7}$$

where the subscript S means that the derivative is evaluated at the surface.

For constant h:

$$\int_S N_i \, k \frac{\partial T}{\partial n} \, dS = h \int_S N_i (\theta - T) \, dS. \tag{8}$$

In what follows the fluid temperature θ is taken to be a specified function of position and time. For purposes of discretization, the fluid temperature is specified at a discrete number of fluid "nodes." If θ_j represents the fluid temperature at fluid node j, then the fluid temperature along the boundary may be represented by

$$\theta = \Sigma N_j \theta_j, \tag{9}$$

where the N_j are one-dimensional forms of the shape functions used for the solid. Substitution in Eq. 8 gives

$$\int_S N_i \, k \, \frac{\partial T}{\partial n} \, dS = \sum_j y^*_{ij} (\theta_j - T_j), \tag{10}$$

where the summation extends over the surface nodes and the coefficients y^*_{ij} are given by

$$y^*_{ij} = h \int_S N_i N_j \, dS. \tag{11}$$

Assembling the contributions from Eq. 10, the subvector $v^{(2)}$ for the convection boundary condition may be written

$$v^{(2)} = \underline{\underline{Y}}^* \underline{\theta}. \tag{12}$$

The contributions $-\Sigma y^*_{ij} T_j$ from Eq. 10 are included by augmenting the matrix $\underline{\underline{Y}}$ (see Eq. 13 below).

3. Constant Temperature

If θ represents the constant temperature desired at the wetted surface, then we can use the convection boundary condition and replace h by a big number (say 10^{20}). Since h is very large, then for thermal equilibrium the temperature

T at the surface will be forced to equal θ. So the subvector $\underline{v}^{(3)}$ for the portion corresponding to the constant temperature boundary condition can be obtained from Eq. 12.

Upon the application of these boundary conditions in a single problem, the right-hand side vector $\underline{v}$ will be combined from the corresponding subvectors and the finite element discretized equation becomes

$$\underline{\underline{C}}\,\dot{\underline{T}} + \underline{\underline{Y}}^{+}\,\underline{T} = \underline{v}, \tag{13}$$

where $\underline{\underline{Y}}^{+} = \underline{\underline{Y}} + \underline{\underline{Y}}^{*}$.

C. EXACT TIME SOLUTION WITH SPATIAL DISCRETIZATION

We consider only the solution of Eq. 13 for $\underline{v}$ = constant with $\underline{T} = \underline{a}$ at time $\tau = 0$. Let $\underline{T}_s$ be a particular solution (steady state) with $\dot{\underline{T}}_s = 0$ so that

$$\underline{T}_s = (\underline{\underline{Y}}^{+})^{-1}\,\underline{v}. \tag{14}$$

For the homogeneous equation

$$\underline{\underline{C}}\,\dot{\underline{T}} + \underline{\underline{Y}}^{+}\,\underline{T} = 0 \tag{15}$$

the assumption $\underline{T} = \underline{w}\exp(-\lambda\tau)$, where $\underline{w}$ is a vector and λ is a scalar, yields the form

$$\underline{\underline{Y}}^{+}\,\underline{w} = \lambda\,\underline{\underline{C}}\,\underline{w}\ . \tag{16}$$

It is apparent that Eq. 16 defines an eigenvalue problem. Let $\underline{\underline{\Lambda}}$ be the spectral matrix and $\underline{\underline{W}}$ the modal matrix with normalization according to

$$\underline{\underline{W}}^{T}\,\underline{\underline{C}}\,\underline{\underline{W}} = \underline{\underline{I}}\ , \tag{17}$$

where $\underline{\underline{I}}$ is the identity matrix of the same order as $\underline{\underline{C}}$. Now let

$$\underline{T} = \underline{\underline{W}} \exp(-\underline{\underline{\Lambda}}\tau)\ \underline{b} \tag{18}$$

where $\underline{b}$ is a constant vector. Substituting this in Eq. 15 gives

$$\underline{\underline{Y}}^{+}\ \underline{\underline{W}} \exp(-\underline{\underline{\Lambda}}\tau)\ \underline{b} = \underline{\underline{C}}\ \underline{\underline{W}}\ \underline{\underline{\Lambda}} \exp(-\underline{\underline{\Lambda}}\tau)\ \underline{b}. \tag{19}$$

Now Eq. 19 is satisfied for all $\underline{b}$ if

$$\underline{\underline{W}}^{T}\ \underline{\underline{Y}}^{+}\ \underline{\underline{W}} = \underline{\underline{\Lambda}}\ , \tag{19'}$$

and this is guaranteed to be satisfied since $\underline{\underline{\Lambda}}$ and $\underline{\underline{W}}$ are spectral and modal matrices for the eigenvalue problem of Eq. 16 with $\underline{\underline{W}}$ normalized according to Eq. 17.

Returning to the original problem (Eq. 13), the complete solution may be written as

$$\underline{T} = \underline{\underline{W}}\ (\underline{\beta} + \exp(-\underline{\underline{\Lambda}}\tau)\ \underline{b}) \tag{20}$$

where

$$\underline{T}_{S} = \underline{\underline{W}}\ \underline{\beta}\ , \tag{21}$$

and $\underline{\beta}$ is a constant vector. Now $\underline{\beta}$ may be found (using Eq. 14) to be

$$\underline{\beta} = \underline{\underline{\Lambda}}^{-1}\ \underline{\underline{W}}^{T}\underline{v}\ . \tag{22}$$

Substituting this result into Eq. 20 and using the initial condition gives the result

$$\underline{b} = \underline{\underline{W}}^{-1}\ \underline{a} - \underline{\underline{\Lambda}}^{-1}\ \underline{\underline{W}}^{T}\ \underline{v}\ . \tag{23}$$

The general solution of Eq. 13 may thus be written as

$$\underline{T} = \underline{\underline{W}}(\underline{\underline{I}} - \exp(-\underline{\underline{\Lambda}}\tau))\ \underline{\underline{\Lambda}}^{-1}\ \underline{\underline{W}}^{T}\underline{v} + \underline{\underline{W}}\ \exp(-\underline{\underline{\Lambda}}\tau)\underline{\underline{W}}^{-1}\underline{a}. \quad (24)$$

For purposes of the present program non-zero components of vector $\underline{v}$ are to be specified as piecewise linear functions of time. During each segment of time history of $\underline{v}$ an analytical solution of Eq. 13 is possible in the form of a particular solution plus a complementary solution such as Eq. 20. At each node the corresponding time variation of temperature will consist of a linear part contributed by the particular solution and a sum of n terms representing the complementary part. Each of these n terms decays exponentially with a separate time constant. In principle it is a straightforward process to find each particular solution and accompanying complementary solution.

Contemplated problems may typically have from 10 to 40 segments required to represent the piecewise linear variation of $\underline{v}$. The number of body nodes n will be of the order of 100 or more. In view of the number of particular solutions required, each accompanied by an individual complementary solution of the form given by Eq. 20, it was concluded that a numerical solution of Eq. 13 would be considerably more economical than an analytic one such as that given by Eq. 24.

D. TIME INTEGRATION

In this section the relative merits of the Runge-Kutta and trapezoidal methods of time integration are discussed. Since either of these methods will give an exact result if the solution is a linear function of time, investigation

is confined to performance on a single scalar equation

$$\dot{y} + \lambda y = 0 \tag{25}$$

whose solution $y = y_o \exp(-\lambda\tau)$ is of the same form as the components of the complementary solution (Eq. 20).

1. Runge-Kutta Method

A method introduced by Runge and subsequently elaborated by Heun and Kutta [3] is widely used for the numerical solution of first order ordinary differential equations. This algorithm prescribes a sequence of calculations for determining the ordinate y_{i+1} at time $\tau_{i+1} = \tau_i + \Delta\tau$ in terms of y_i and values of $\dot{y}$ at intermediate and end points of the interval $\Delta\tau$. The fourth-order form, which requires four evaluations of $\dot{y}$, gives for Eq. 25 the result

$$\frac{y_{i+1}}{y_i} = 1 - \lambda\Delta\tau + \frac{(\lambda\Delta\tau)^2}{2!} - \frac{(\lambda\Delta\tau)^3}{3!} + \frac{(\lambda\Delta\tau)^4}{4!} . \tag{26}$$

The right-hand side of Eq. 26 represents the first five terms of the Taylor expansion of the exact solution $(y_{i+1}/y_i = \exp(-\lambda\Delta\tau))$ so we may conclude that the relative error in each time step is less than modulus of the next term: $(\lambda\Delta\tau)^5/5!$.

In addition to providing the apparent prospect for high precision indicated by this error bound, the Runge-Kutta method also permits changes of time increment during the integration process without requiring additional computationally expensive matrix decompositions. The attractiveness of these two features dictated a thorough exploration of the potential of this method for the present application. The

disqualifying defect which emerged after studying a number of examples is readily appreciated from examination of Table I. For values of $\lambda\Delta\tau$ less than 0.5 it is apparent that Runge-Kutta scheme affords acceptable engineering accuracy. However, when the method is applied to solution of Eq. 13 we must deal with a number of λ's equal to the number of nodes (see Eq. 20). This number may be greater than 100 and the ratio of the largest λ to the smallest may easily exceed 1000. Although the solution is dominated by the contributions of the eigenvectors corresponding to the smaller λs, it is clear that the solution will be unstable if the largest $\lambda\Delta\tau$ exceeds about 2.7. Because an unacceptably small $\Delta\tau$ is required in typical problems, the Runge-Kutta method was rejected.

2. Trapezoidal Method

The trapezoidal method estimates y_{i+1} from the formula

$$y_{i+1} = y_i + \frac{\Delta\tau}{2}(\dot{y}_i + \dot{y}_{i+1}). \tag{27}$$

Substituting for $\dot{y}_i$ and $\dot{y}_{i+1}$ from Eq. 25 and rearranging gives

$$\frac{y_{i+1}}{y_i} = \frac{2 - \lambda\Delta\tau}{2 + \lambda\Delta\tau}. \tag{28}$$

Series expansion of the right-hand side provides an error bound (per step):

$$(\lambda\Delta\tau)^3/12.$$

From the point of view of the size of $\lambda\Delta\tau$ this method has no stability limit, but has slow attenuation with alteration

TABLE I

ATTENUATION FACTOR COMPARISON

Fourth Order Runge-Kutta Algorithm

$\lambda\Delta\tau$	y_{i+1}/y_i (Runge-Kutta)	$\exp(-\lambda\Delta\tau)$ (Exact)
.0001	.9999	.9999
.001	.9990	.9990
.01	.9901	.9901
.1	.9048	.9048
.2	.8187	.8187
.5	.6068	.6065
1.0	.3750	.3679
2.0	.3333	.1353
2.5	.6484	.0821
3.0	1.3750	.0498
4.0	5.0000	.0183
8.0	110.3333	.0003
10.0	291.0000	.0000
20.0	5514.3333	.0000
50.0	240784.3333	.0000
100.0	4004901.0000	.0000

in sign for large $\lambda\Delta\tau$. Table II shows this behavior.

Since a wide usable range of $\lambda\Delta\tau$ is essential and the stability of trapezoidal integration is guaranteed, this method is chosen for the present program.

Applying the trapezoidal algorithm to Eq. 13 yields

$$\underline{\underline{A}}\ \underline{T}^{i+1} = \underline{\underline{G}}\ \underline{T}^{i} + \frac{\Delta\tau}{2}\ (\underline{v}^{i+1} + \underline{v}^{i}), \tag{29}$$

where

$$\underline{\underline{A}} = \underline{\underline{C}} + \frac{\Delta\tau}{2}\ \underline{\underline{Y}}^{+},$$

$$\underline{\underline{G}} = \underline{\underline{C}} - \frac{\Delta\tau}{2}\ \underline{\underline{Y}}^{+}.$$

and the superscripts denote evaluation at discrete time intervals $\Delta\tau$. If m is the order of the capacitance and admittance matrices, $\underline{\underline{C}}$ and $\underline{\underline{Y}}$, then once a certain step size $\Delta\tau$ is chosen, it requires $m^3/3$ operations to perform the needed triangular decomposition of $\underline{\underline{A}}$. Thus, for large m, a change of step size $\Delta\tau$ becomes costly from the point of view of computer time. Accordingly, in the present program only one time step size is used throughout each problem.

Also, for assuring sufficiently rapid attenuation of the components corresponding to the large $\lambda\Delta\tau$, the following correction is utilized.

2. Irons' Correction

Irons proposed a scheme [4] for augmenting the attenuation of the contributions of those eigenvectors for which $\lambda\Delta\tau$ is large. Define

TABLE II

ATTENUATION FACTOR COMPARISON

Trapezoidal Integration

$\lambda\Delta\tau$	y_{i+1}/y_i (Trapezoidal)	$e^{-\lambda\Delta\tau}$ (Exact)
.0001	.9999	.9999
.001	.9990	.9990
.01	.9901	.9901
.1	.9048	.9048
.2	.8182	.8187
.3	.7391	.7408
.5	.6000	.6065
1.0	.3333	.3679
2.0	.0000	.1353
2.5	-.1111	.0821
3.0	-.2000	.0498
4.0	-.3333	.0183
8.0	-.6000	.0003
10.0	-.6667	.0000
20.0	-.8182	.0000
50.0	-.9231	.0000
100.0	-.9608	.0000

$$y_i^* = .25\ y_{i-1} + .5\ y_i + .25\ y_{i+1}, \qquad (30)$$

where y_i and y_{i+1} are obtained from y_{i-1} by trapezoidal integration.

In the program presented in Appendix B, Eq. 30 is used after every 10 steps of time integration. Table III shows the resulting modifications.

E. ESTIMATION OF EXTREME EIGENVALUES

Analytic results for one-dimensional heat conduction give, for an eigenvalue,

$$\lambda = \frac{\pi^2 k}{4\rho c}\ \frac{1}{d^2}\ , \qquad (31)$$

where d is the distance between points of extreme temperature and zero temperature.

If we use this to estimate the smallest λ in cylindrical coordinates, two modifications are recommended.

1. Assume that the point of zero temperature is in the fluid at a distance from the wall equal to k/h, where h is the surface heat transfer coefficient.
2. If there are two approximately orthogonal paths for heat flow from the (single) maximum temperature point, then replace $1/d^2$ in the above formula by

$$\frac{1}{d^2} = \frac{1}{d_{min.}^2} + \frac{1}{d_{max.}^2}\ . \qquad (32)$$

For estimating the largest λ, the surface heat transfer coefficient has no significant effect. We may continue to

TABLE III

EFFECTS OF USING IRONS' CORRECTION
AFTER 10 STEPS OF INTEGRATION

$\lambda\Delta\tau$	$\exp(-10\lambda\Delta\tau)$ Exact	y_{10}/y_0 Trapezoidal	y^*_{10}/y_0 Corrected
0.00010	0.99900	0.99900	0.99900
0.00020	0.99800	0.99800	0.99800
0.00100	0.99005	0.99005	0.99005
0.01000	0.90484	0.90484	0.90486
0.10000	0.36788	0.36757	0.36849
0.20000	0.13534	0.13443	0.13579
0.30000	0.04979	0.04866	0.04978
0.50000	0.00674	0.00605	0.00645
1.00000	0.00005	0.00002	0.00002
2.00000	0.00000	0.00000	0.00000
4.00000	0.00000	0.00002	-0.00001
8.00000	0.00000	0.00605	-0.00040
10.00000	0.00000	0.01734	-0.00072
20.00000	0.00000	0.13443	-0.00136
50.00000	0.00000	0.44914	-0.00072
100.00000	0.00000	0.67028	-0.00027

use the same formula for $\frac{1}{d^2}$, but now consider only the smallest element and take

$$d_{min} = \frac{\text{length of smallest side}}{4}, \qquad d_{max} = \frac{\text{length of largest side}}{4}. \tag{32'}$$

A comparison of estimates based on Eq. 31, 32, 32' with the exact solution of the eigenvalue problem has been carried out for several examples. Based on these comparisons, it is believed that these estimates are sufficiently accurate for choosing a time step and estimating the time of occurrence of the extreme stresses.

III. ONE-DIMENSIONAL HEAT CONDUCTION

For comparison of numerical (time) integration methods, studies of one-dimensional heat conduction were made. In this section numerical results for trapezoidal time integration using Irons' correction are compared with the exact transient temperature solution.

Consider a flat slab of thickness L with conductivity k, density ρ, specific heat c, zero initial temperature, one face insulated and the other in contact with fluid at temperature T_f (Fig. 1). The surface heat transfer is h. The exact transient heat conduction solution is available [5] as

$$T = T_f \left\{ 1 - 2 \sum_{i=1}^{\infty} \frac{\mathrm{Sin}\mu_i L \, \mathrm{Cos}\mu_i x}{\mu_i L + \mathrm{Sin}\mu_i L \, \mathrm{Cos}\mu_i L} \cdot e^{-\mu_i^2 \frac{k}{\rho c} \tau} \right\} \quad (33)$$

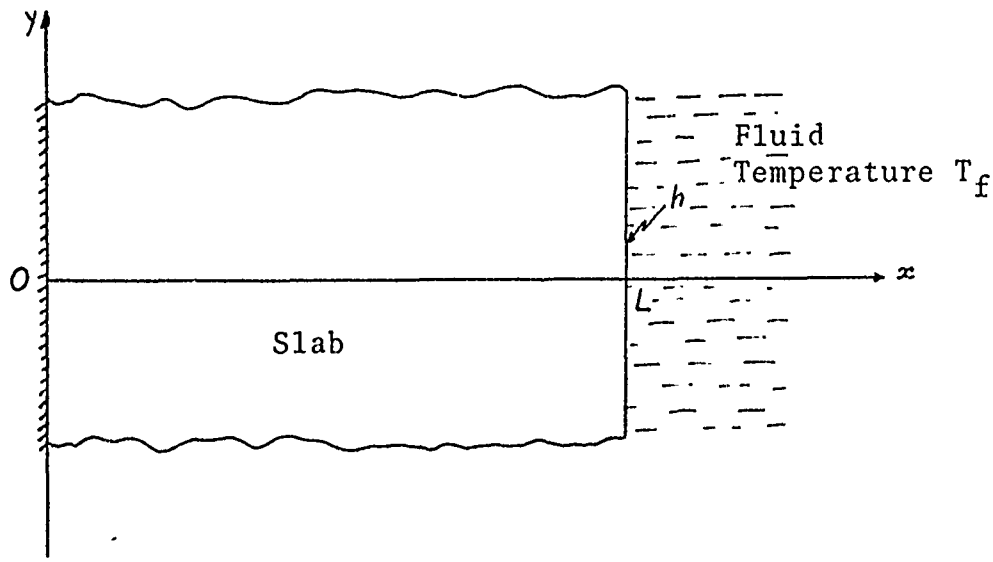


Figure 1. Slab with One Face Insulated and Another in Contact with Fluid.

where $k/\rho c$ is the diffusivity and the μ_i are the solutions of the transcendental equation

$$\text{Tan } \mu L = \frac{hL}{k} \frac{1}{\mu L} . \qquad (34)$$

For the finite element comparisons we subdivide the distance L into m one-dimensional 3-noded elements and use the corresponding isoparametric shape functions. (The working equations, shape functions and the element capacitance and admittance matrices are given in Appendix A.)

In the course of this investigation separate computer programs were written to evaluate nodal transient temperatures using the following methods:

(a) exact transient temperature solution, Eq. 33;

(b) exact time solution with spatial discretization (section II.C, Eq. 24);

(c) Runge-Kutta time integration (section II.D, part a);

(d) trapezoidal time integration (section II.D, part b);

(e) trapezoidal time integration with Irons' correction (section II.D, part c).

For an initial step change of fluid temperature from zero to 1, transient temperatures have been found. For these comparisons the parameters (in consistent units) were taken to be:

$$L = 8,\ \rho = 25,\ k = 8,\ c = 5 \text{ and } h = 5$$

The distance L was subdivided into $m = 3$ elements. For the present purpose, comparison is confined to the exact solution (item (a)), and the finally adopted system (item (e)).

In Table IV, temperatures at the two faces (x = 0, x = 8) are compared for various times. The trapezoidal integration has been performed using the constant time increment unity and the Irons' correction is applied after every 10 increments.

It is believed that Table IV demonstrates that the numerical integration method gives adequate accuracy for engineering applications.

TABLE IV

COMPARISON OF ONE-DIMENSIONAL TRANSIENT TEMPERATURES

Method	x	Time=0	Time=10	Time=20	Time=30	Time=50	Time=70
Exact	0.	.00000	.00000	.00000	.00002	.00095	.00560
Trapezoidal with Irons'✓	0.	.00000	.00028	.00007	.00066	.00207	.00619
Exact	8.	.00000	.38431	.47684	.53284	.60264	.64685
Trapezoidal with Irons'✓	8.	.00000	.38720	.47716	.53262	.60256	.64686

✓Irons' correction is used after every 10 steps of trapezoidal integration.

IV. STRESS PROBLEM

For bodies of revolution deformed symmetrically under axisymmetric loading, the stress distribution is two-dimensional. Since deformation is symmetric about the axis of revolution, cylindrical coordinates (R,Z,θ) are used. It follows that the stress components are independent of the angle θ and all derivatives with respect to θ are zero. Also the components of shearing stress $\tau_{R\theta}$ and $\tau_{Z\theta}$ vanish on account of the symmetry. But since any radial displacement induces a strain ε_θ in the circumferential direction, this non-zero component of strain and the three in-plane components $(\varepsilon_Z, \varepsilon_R, \gamma_{RZ})$, complete the state of strain at a point in any axisymmetric situation. Hence the state of stress for an axisymmetric body under axisymmetric loading is given by

$$\underline{\sigma} = \langle \sigma_Z \ \sigma_R \ \sigma_\theta \ \tau_{RZ} \rangle^T . \tag{35}$$

In this chapter the stiffness matrix of an axisymmetric body and the thermal, pressure, and centrifugal load vectors are formulated and, finally, evaluation of stresses at a point is discussed. The treatment closely follows that of Zienkiewicz [1] and this reference should be consulted for further details.

A. STIFFNESS MATRIX

The elements used are bodies of revolution (about the Z axis). For analysis it is sufficient to describe the cross-section in the R,Z plane. In Fig. 2 such an element and the local ξ,η coordinates are shown.

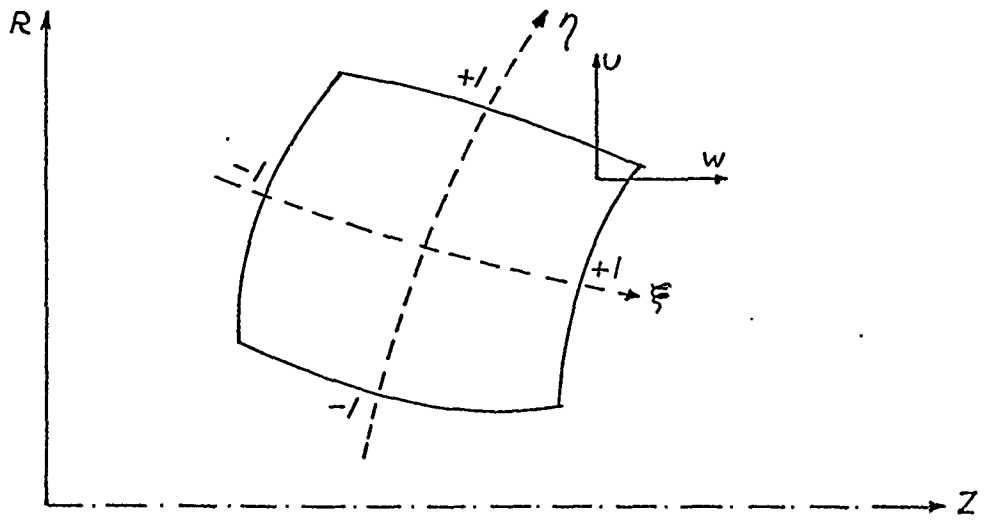


Figure 2. Quadrilateral Element Representation.

If u and w are the displacement components at a point in the directions of R and Z respectively, then these displacement components may be defined in terms of the nodal displacements by the appropriate isoparametric shape functions as

$$u = \sum_{i=1}^{8} N_i u_i \ , \quad w = \sum_{i=1}^{8} N_i w_i \ , \tag{36}$$

where N_i, a function of ξ and η , is the shape function for element node i and u_i, w_i, are the nodal displacement components. The strain-displacement relations [6] can now be used to obtain the components of the strain vector. Thus

$$\underline{\varepsilon} = \underline{\underline{B}}\ \underline{\delta}\ , \tag{37}$$

where $\underline{\underline{B}}$ is the standard rectangular strain-displacement matrix of any finite element formulation, a function of the local coordinates ξ and η, and $\underline{\delta}$ is the vector of nodal displacement. (See Appendix A, part 3, where the applicable formulas and useful equations are summarized).

If the elasticity matrix for an isotropic material is $\underline{\underline{D}}$, then the stress vector $\underline{\sigma}$ at a point is given by

$$\underline{\sigma} = \underline{\underline{D}}\ \underline{\varepsilon}\ . \tag{38}$$

Now, by evaluation of the total strain energy in the element, the element stiffness matrix can readily be obtained as

$$\underline{\underline{K}}^e = \int \underline{\underline{B}}^T\ \underline{\underline{D}}\ \underline{\underline{B}}\ dV\ , \tag{39}$$

where the integration extends over the volume of the element.

In the present program the upper triangle of each element stiffness matrix is evaluated by numerical integration using four Gauss points within the range of ξ and of η [1]. The element contribution is immediately placed in the system stiffness matrix, which is stored in banded form.

B. THERMAL LOAD VECTOR

If we denote T as the difference between local temperature and reference temperature, then the thermal strain $\underline{\varepsilon}_o$ is given as

$$\underline{\varepsilon}_o = \underline{U}\alpha T, \tag{40}$$

where

$$\underline{U} = \langle 1\ 1\ 1\ 0 \rangle^T$$

and α is the coefficient of thermal expansion.

The thermal load vector $\underline{F}_T^e$ is given by Zienkiewicz [1] as

$$\underline{F}_T^e = \int \underline{\underline{B}}^T \underline{\underline{D}}\ \underline{\varepsilon}_o^e\ dV\ . \tag{41}$$

From the point of view of the numerical evaluation it is interesting to note, however, that the product $\underline{\underline{D}}\ \underline{\varepsilon}_o^e$ in Eq. 41 will reduce to

$$\underline{\underline{D}}\ \underline{\varepsilon}_o^e = \frac{E\alpha T}{1-2\nu}\ \underline{U}\ , \tag{42}$$

where E is the modulus of elasticity and ν is Poisson's ratio. Thus

$$\underline{F}_T^e = \frac{E\alpha}{1-2\nu} \int T\ \underline{\underline{B}}^T\ \underline{U}\ dV\ , \tag{43}$$

where $T = \sum_{i=1}^{n} N_i T_i$ and n is the number of nodes in each element.

In the attached computer program in Appendix B the advantage of the simplicity of Eq. 42 has been utilized. Also, since $\underline{\underline{B}}$ has some zero components, in the process of multiplication of $\underline{\underline{B}}^T\ \underline{U}$, simply the addition of the appropriate non-zero components of each column of the $\underline{\underline{B}}^T$ has been performed. Finally, Eq. 43 has been integrated with four Gauss points.

C. PRESSURE LOAD VECTOR

Consider a quadrilateral element as in Fig. 3 on the boundary of the axisymmetric cross-section where constant pressure P is applied. The infinitesimal force dF due to the normal pressure acting on the inner infinitesimal circumferential surface dS is

$$dF = PdS = 2\pi RP\, d\ell \ , \tag{44}$$

where $d\ell$ is the infinitesimal length along the side of quadrilateral.

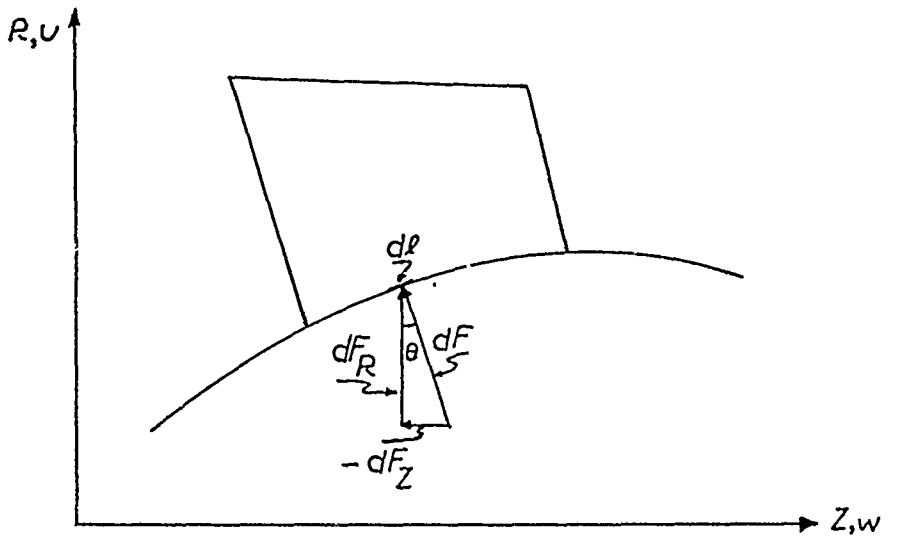


Figure 3. Boundary Element Under Pressure.

Let F_R and F_Z be the components of the pressure force in the R and Z directions respectively, then

$$dF_R = dF \text{ Cos } \theta, \quad dF_Z = -dF \text{ Sin } \theta \ , \tag{45}$$

where $\text{Sin } \theta = \frac{dR}{d\ell}$ and $\text{Cos } \theta = \frac{dZ}{d\ell}$. Therefore

$$dF_R = (2\pi RP)dZ \ , \quad dF_Z = -(2\pi RP)dR. \tag{46}$$

Since the total work W done by the normal force F is equal to the sum of the work done in R and Z directions,

$$W = \int u \, dF_R + \int w \, dF_Z$$

$$= 2\pi P \, (\int R \, u \, dZ - \int R \, w \, dR). \qquad (47)$$

Now, by using the appropriate shape functions, each component of Eq. 47 may be defined in terms of the nodal values. Since

$$W = (\underline{\delta}^e)^T \, \underline{F}_p^e, \qquad (48)$$

where $\underline{F}_p^e$ is the element pressure load vector
vector contributed by node j is explicitly given as

$$F_{j_p}^e = \begin{bmatrix} F_R \\ F_Z \end{bmatrix}_{j_p}^e = 2\pi P \int_{-1}^{1} (\Sigma N_i R_i) \begin{bmatrix} \Sigma \frac{\partial N_i}{\partial \xi} Z_i \\ -\Sigma \frac{\partial N_i}{\partial \xi} R_i \end{bmatrix} N_j \, d\xi \, , \qquad (49)$$

where N_j in Eq. 49 is the appropriate shape function for node j.

D. CENTRIFUGAL LOAD VECTOR

Refer again to Fig. 2 and assume that the body is rotating about the Z axis with angular speed Ω. Then the centrifugal force per unit volume at a point distant R from the Z axis will be $\rho\Omega^2 R$, where ρ is the density of the material. The work done in this case is

$$W = \int_V \rho\Omega^2 \, R \, u \, dV \, . \qquad (50)$$

For constant $\rho\Omega^2$ the corresponding element centrifugal load vector is readily obtained by evaluation of the components of the integral in Eq. 50 in terms of the nodal variables, i.e.,

$$F^e_{j_c} = 2\pi\rho\Omega^2 \int_{-1}^{1}\int_{-1}^{1} (\Sigma N_i R_i)^2 \det \underline{\underline{J}}\, N_j \, d\xi d\eta, \qquad (51)$$

where $\det \underline{\underline{J}}$ is the determinant of the Jacobian coordinate transformation matrix (see Appendix A).

E. STRUCTURAL BOUNDARY CONDITIONS

The structural boundary conditions implemented in the program are:

(a) one or more nodes prevented from moving axially;

(b) one or more nodes prevented from moving radially;

(c) the right-hand end cross-sectional plane remains plane and the transmitted axial force is specified. Hereinafter this will be referred to as the plane-end boundary condition.

The computer program presented in Appendix B has the capability of handling any combination of the above mentioned structural boundary conditions. For boundary conditions of the types (a) and (b), simply multiplying the corresponding diagonal component of the stiffness matrix by 10^{20} gives zero displacement [8] (for practical purposes). For the boundary condition of type (c) both ends are initially fixed axially for all solutions. An additional solution is obtained for unit axial displacement of one end. The axial force is evaluated for each solution. Superposition is performed by adding the displacement vectors for the given

loadings (thermal plus mechanical), plus an appropriate fraction of the vector found for unit axial displacement. This fraction is chosen so that the resultant axial load has the specified value.

F. SYSTEM EQUATION SOLVER

Once the desired structural boundary conditions are applied, then the problem is to find the nodal displacement vector $\underline{\delta}$, corresponding to a given number of load vectors. We have

$$\underline{\underline{K}}\,\underline{\delta} = \underline{F}\ , \tag{52}$$

where $\underline{\underline{K}}$ is the system stiffness matrix in banded form and $\underline{F}$ is a load vector. In the present computer program a single Cholesky decomposition is performed on $\underline{\underline{K}}$. Then, by a process of forward and back substitution, each load vector is replaced by the corresponding displacement vector.

G. PRINCIPLE OF SUPERPOSITION

Upon the evaluation of the displacement vectors due to the various types of loading, the principle of superposition can be applied on the displacement vectors. On each thermal displacement vector the displacement due to any other type of loading is superimposed and, as the result, the number of displacement vectors is reduced to the number of thermal load vectors.

H. STRESS EVALUATION

From the system displacement vector $\underline{\delta}$, the displacement vector of each element $\underline{\delta}^e$ may be obtained easily. Then the

corresponding element strain vector $\underline{\varepsilon}^e$ at any point can be found from

$$\underline{\varepsilon}^e = \underline{\underline{B}}\ \underline{\delta}^e \ . \tag{53}$$

Finally, the corresponding element stress vector $\underline{\sigma}^e$ is obtained by

$$\underline{\sigma}^e = \underline{\underline{D}}\ (\underline{\varepsilon}^e - \underline{\varepsilon}_o^e) \ . \tag{54}$$

Since normally the stresses on the inner and outer surfaces of axisymmetric bodies are desired, in the computer program presented in the Appendix B provision has been made to calculate the stresses at the two Gauss points corresponding to $\xi = \pm \frac{1}{\sqrt{3}}$ on the inner and outer boundaries of each element (where $\eta = \pm 1$).

Upon the evaluation of the stresses at each point the mean stress and the octahedral shearing stress [7] are calculated. The program gives as output the extreme values of these stresses, the R and Z coordinates of the corresponding points, and the times of occurrence.

V. ONE-DIMENSIONAL TRANSIENT STRESSES

In this section a one-dimensional comparison of stresses is made between exact and finite element results. The transient temperature problem is the one previously described in Section III.

If the slab edges are free to translate in the plane of the slab, but are prevented from rotating, the exact solution for thermal stress [9] is

$$\sigma_y = \sigma_z = \frac{E\alpha}{1-\nu} (T_{avg} - T) , \qquad (55)$$

where T is the local temperature (Eq. 33), and

T_{avg} is the average temperature in the slab

If we choose E = 2, α = .50, and ν = 0 (all in consistent units), then the maximum stress obtained from the exact solution is

$$\sigma_{max} = -.477786$$

and it occurs at x = 8, τ = 73.

Using the finite element technique with trapezoidal time integration and Irons' correction every 10 steps, the maximum stress is found to be

$$\sigma_{max} = -.477778$$

and it also occurs at x = 8, τ = 73, as before.

It is observed that the method chosen gives excellent results.

VI. TEST PROBLEMS

Program integrity and accuracy have been verified by solving a number of test problems. Since stresses, whose evaluation depends upon derivatives of displacements, are known to be less accurate than temperatures, comparisons with exact results are confined to stresses. Individual problems are described below.

A. THICK CYLINDER

Consider a thick cylinder with inside radius 30 inches and outside radius 50 inches and the following material properties.

Modulus of elasticity	$E = 28.9 \times 10^6$	Psi
Poisson's ratio	$\nu = .28$	-
Coefficient of thermal expansion	$\alpha = 7.22 \times 10^{-6}$	1/F°
Thermal conductivity	$k = 28.$	$\frac{\text{Btu}}{\text{hr.ft°F}}$
Density	$\rho = 489.$	Lbm/ft^3
Specific heat	$c = .111$	$\frac{\text{Btu}}{\text{Lbm.°F}}$

An arbitrary length of 25 inches has been selected for the cylinder and it has been subdivided into two different element representations as in Figs. 4 and 5. The plane-end boundary condition with zero axial force is used. The stresses for various types of loading are compared with the corresponding exact analytic solutions as described below.

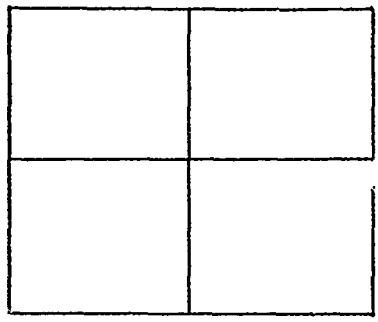


Figure 4. Two Radial Elements Representation of Thick Cylinder.

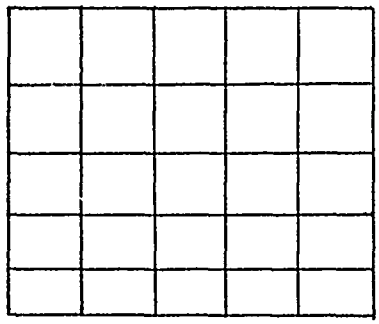


Figure 5. Five Radial Element Representation of Thick Cylinder.

1. Thermal Loading

For a linear variation of the temperature T = 20R the stresses obtained by the finite element method for the above representations are compared with the exact analytic solution in Fig. 6. The τ_{RZ} for this problem clearly is zero and the one obtained by the program was 10^{-9}. The accuracy of the other results is clearly satisfactory.

2. Pressure Loading

A uniform pressure of 1000 psi. acts on the inner surface of the cylinder. Again, τ_{RZ} is zero and the program gives 10^{-10}. In Fig. 7 the other stresses induced by this uniform pressure are compared with the exact solutions. Here also the accuracy of the results, even with two radial elements, is adequate.

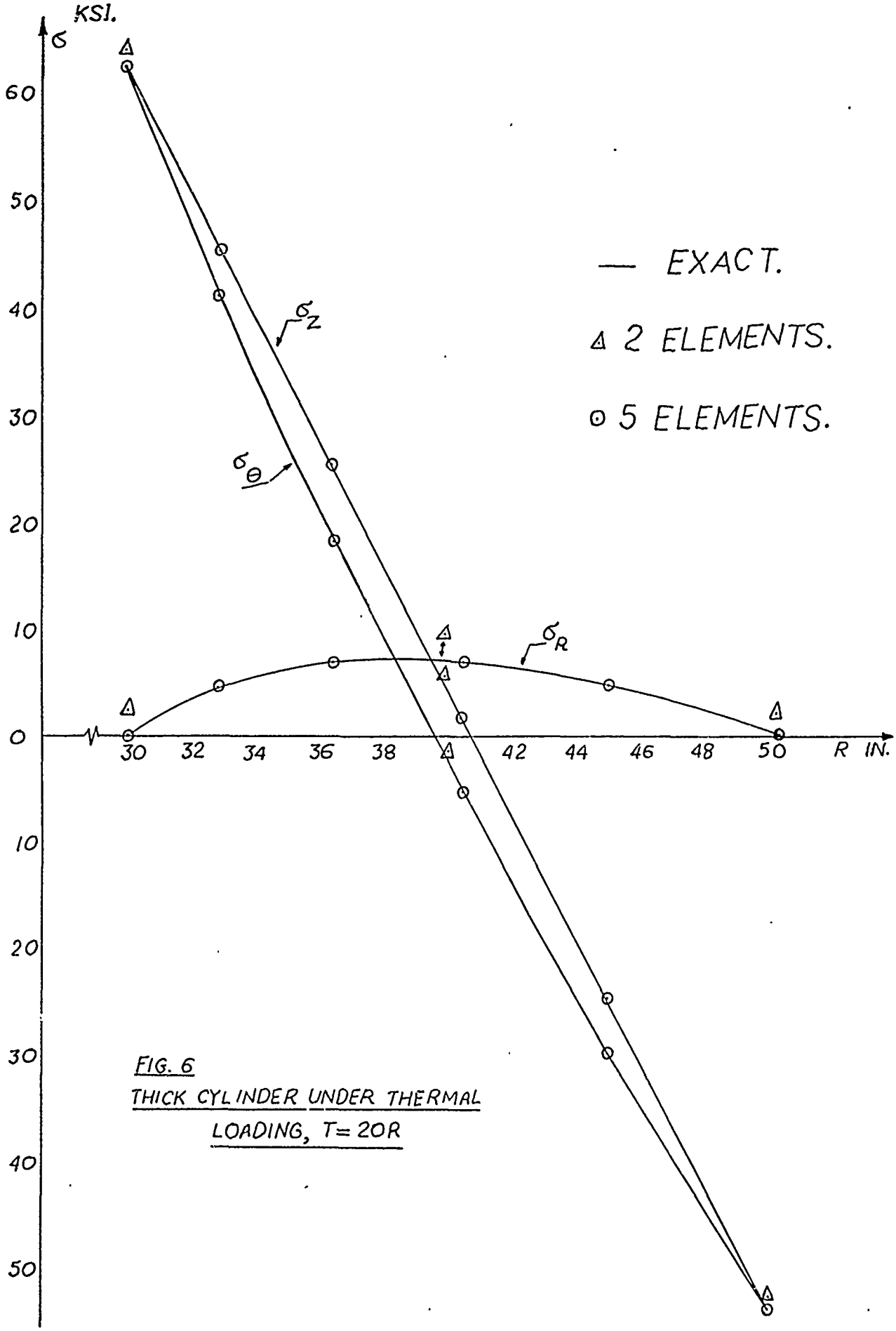


FIG. 6
THICK CYLINDER UNDER THERMAL LOADING, T= 20R

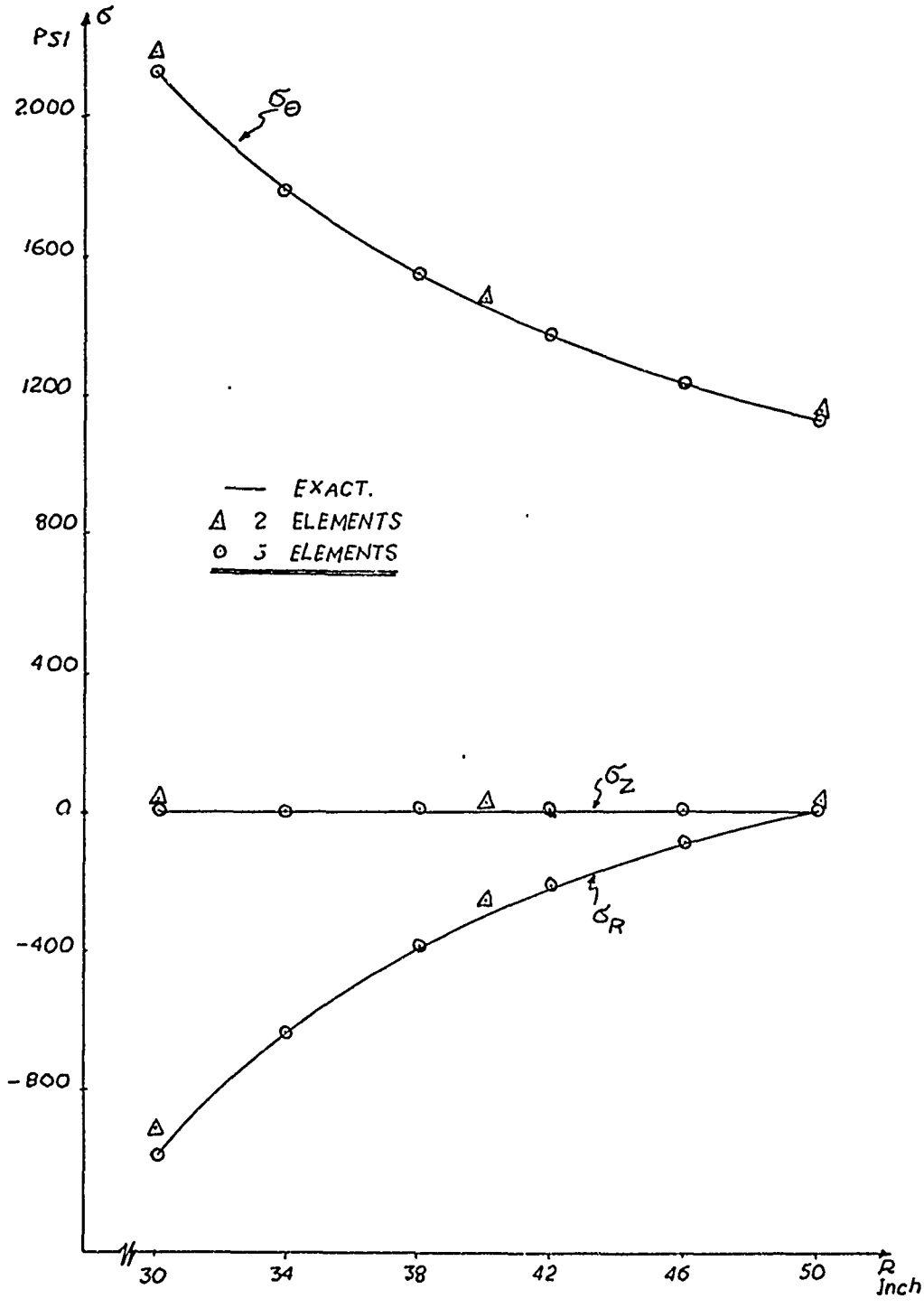


FIG. 7 THICK CYLINDER UNDER INTERNAL PRESSURE.

3. Centrifugal Loading

The speed of rotation has been assumed to be 500 revolutions per minute. The stresses obtained by the program are compared with the analytic solution in Fig. 8. In this case also the results obtained by the program, even with only two radial elements, are very close to the exact solution.

B. HOLLOW SPHERE

We consider a hollow sphere with the same material properties as in the thick cylinder with inside spherical radius 30 inches and outside 50 inches. The loading is thermal with T = 20 ·(spherical radius). Symmetry permits using only half of the sphere for the computer analysis. The elements representation is given in Fig. 9. Since the program gives stresses in cylindrical coordinates, these have been transformed to the spherical coordinates for comparison with the exact solution in Fig. 10. The accuracy of the results is noteworthy.

C. THERMAL STRESSES IN NOZZLE

This problem concerns thermal stresses near the intersection of a cylindrical pipe and the spherical vessel. Fig. 11 gives the cross-section of the structure which is to be analysed. The material properties are:

Modulus of elasticity	$E = 29.3 \times 10^{6}$	Psi
Poisson's ratio	$\nu = .30$	-
Thermal expansion coefficient	$\alpha = 7.6 \times 10^{-6}$	1/F°

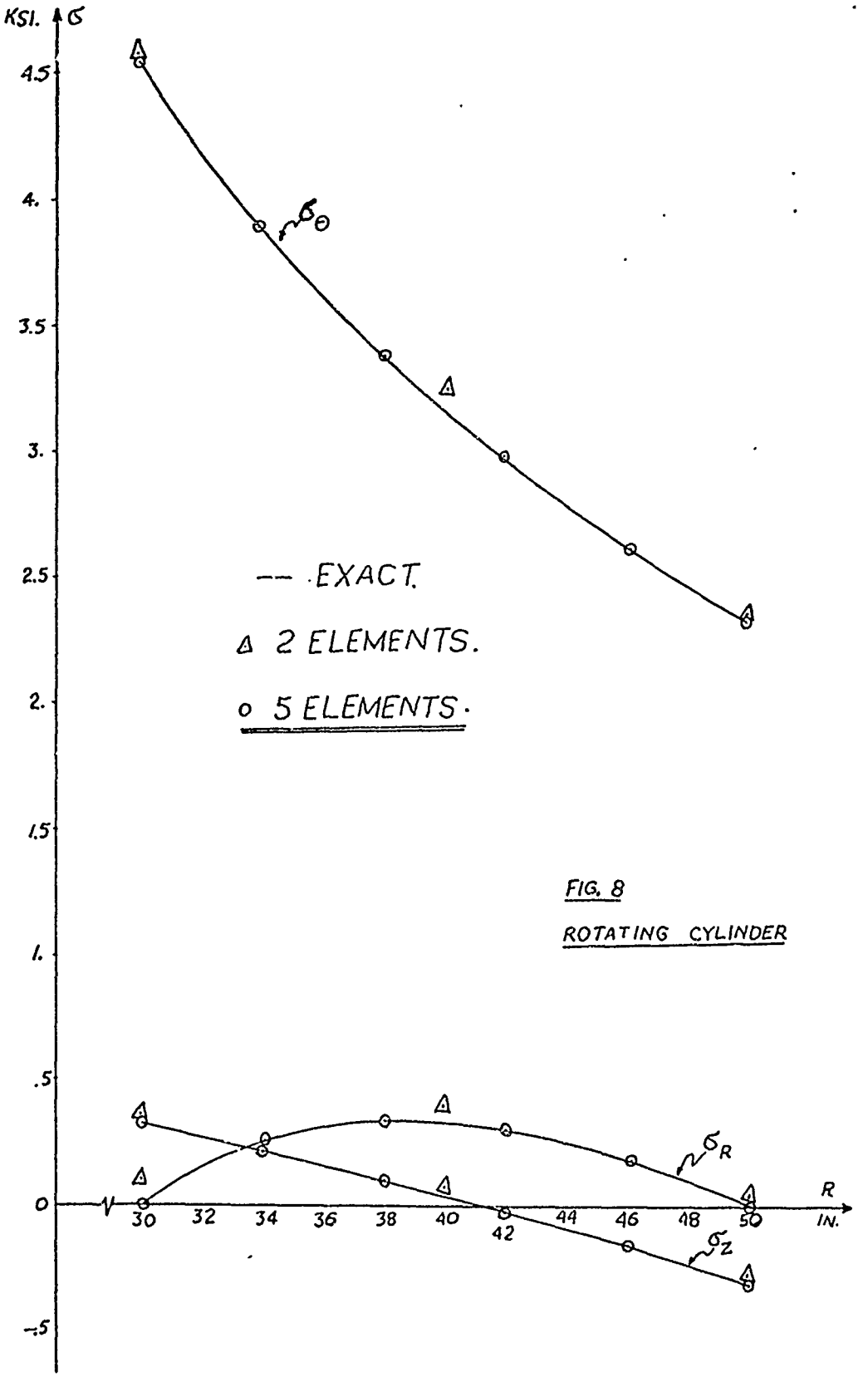


FIG. 8

ROTATING CYLINDER

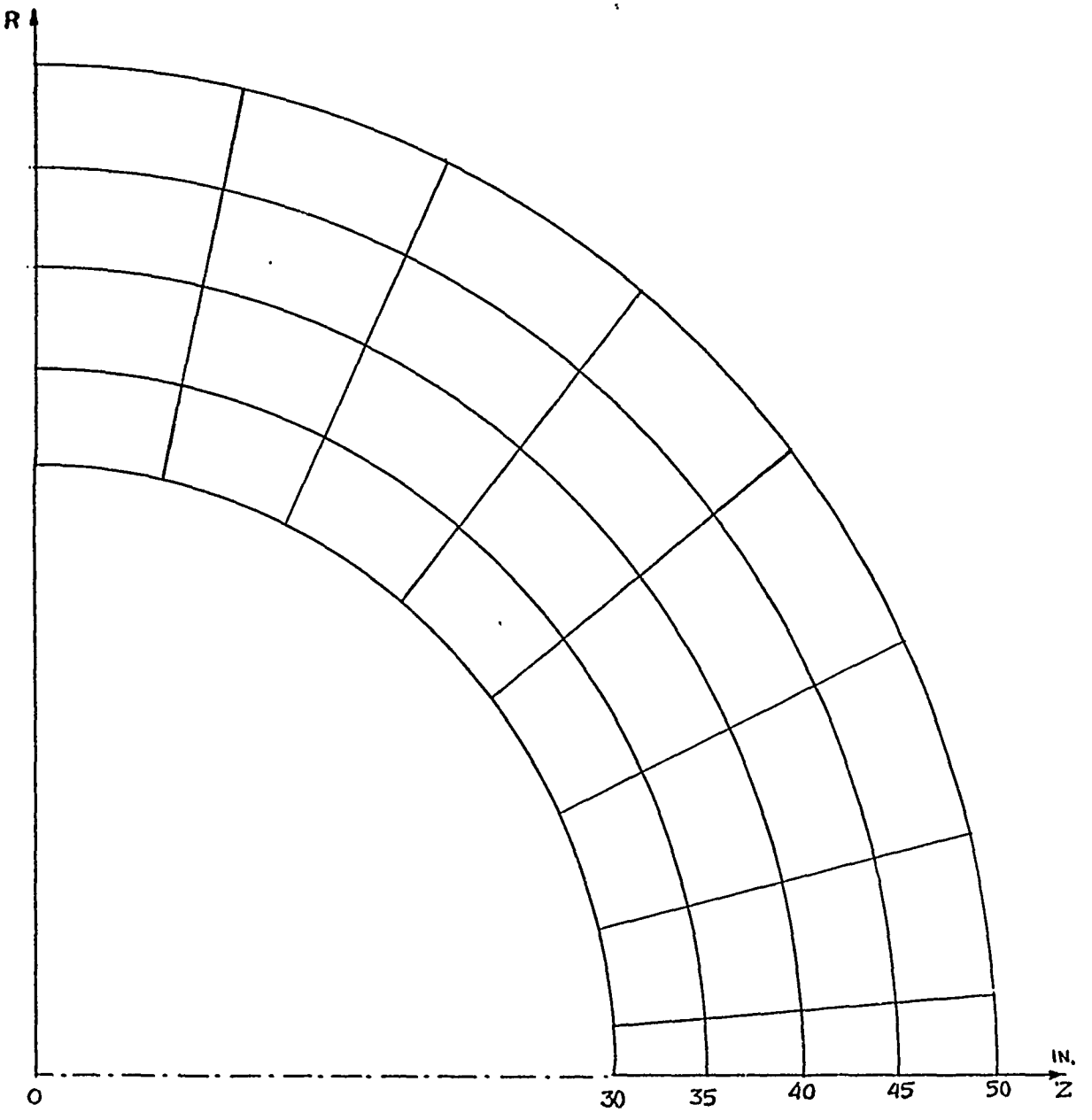


Figure 9. Hollow Semi-sphere 32 Elements Representation.

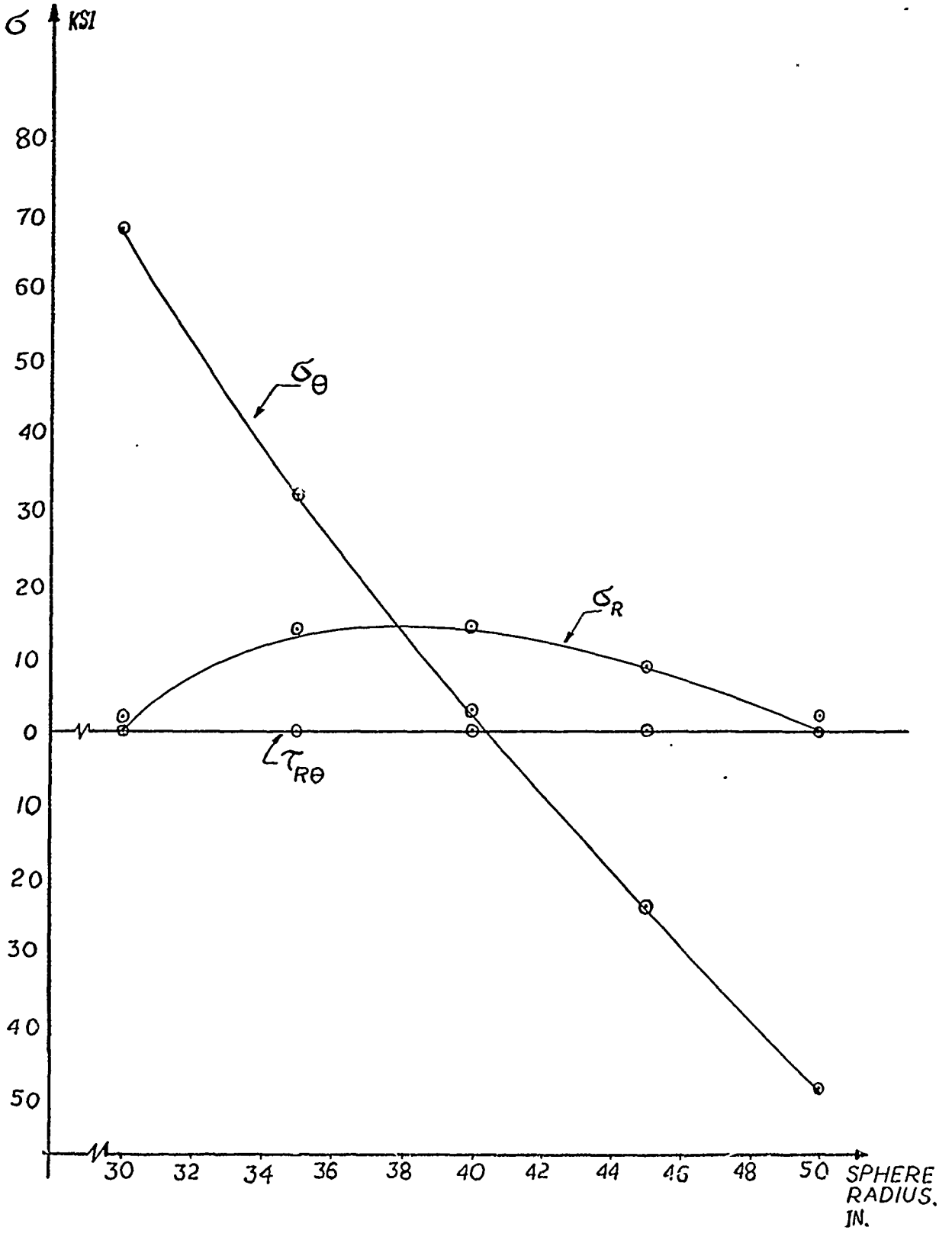
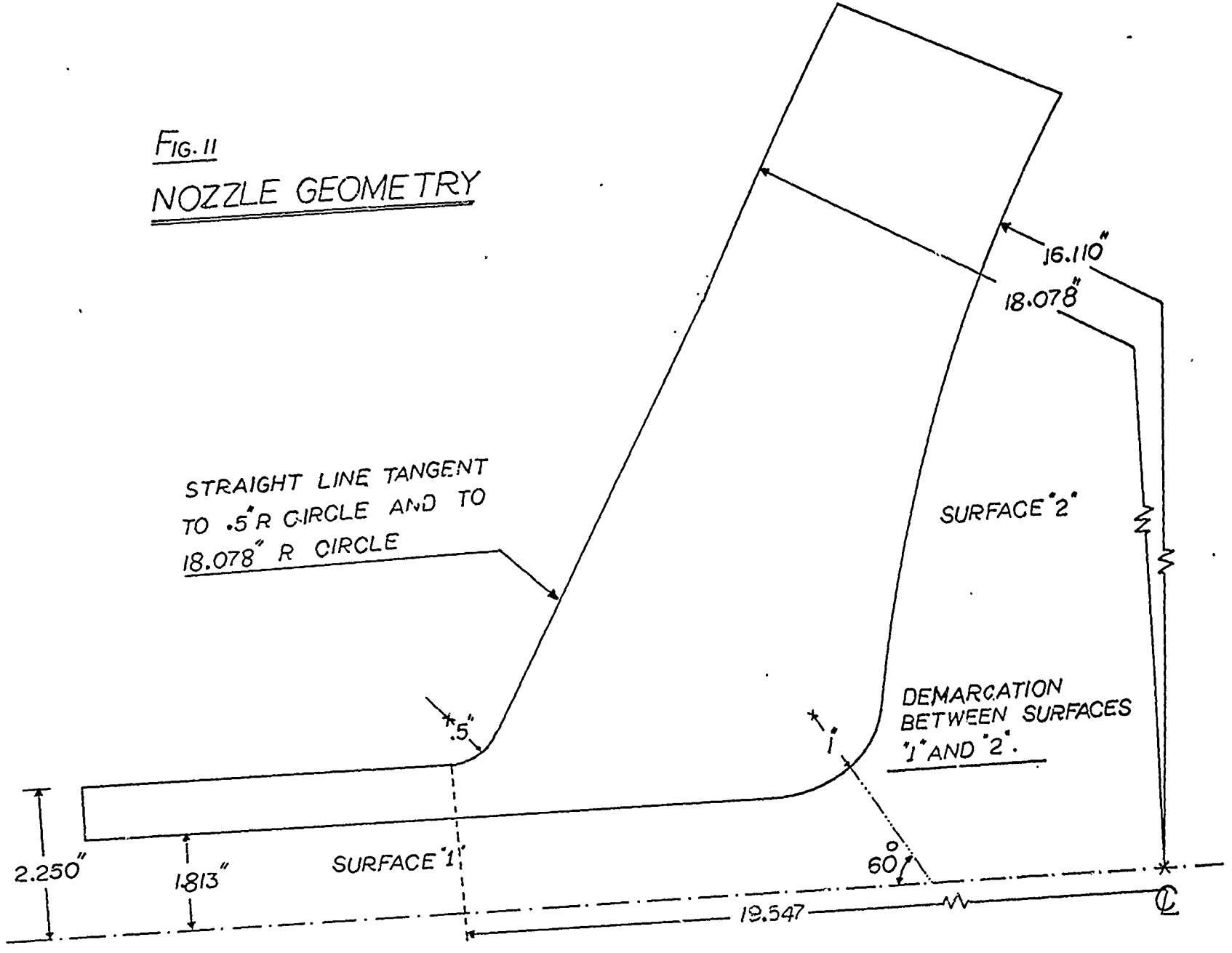


Figure 10. Thermal Stresses in Hollow Sphere
T = 20 (spherical radius).

FIG. 11

NOZZLE GEOMETRY



Thermal conductivity	k = 10.25	Btu/ft.hr.°F
Density	ρ = 530.5	Lbm/ft^3
Specific heat	c = .128	Btu/Lbm°F

The loading results from the thermal transient in the fluid contained in the nozzle. This fluid is in contact with the structure on surface "1" (Fig. 11) and has the entry temperature time variation as in Fig. 12. The fluid in the sphere, which is in contact with the structure on surface "2" (Fig. 11), has the constant temperature 478°F. At $\tau = 0$ structure has a uniform temperature of 478°F and is stress free. Exterior surface of the structure is insulated. The flow velocity past surface 1 is 8.5 ft/sec (inward) and there is no flow past surface 2. The surface heat transfer coefficients are 1393 and 2910 Btu/hr.ft^2°F for surfaces "1" and "2" respectively.

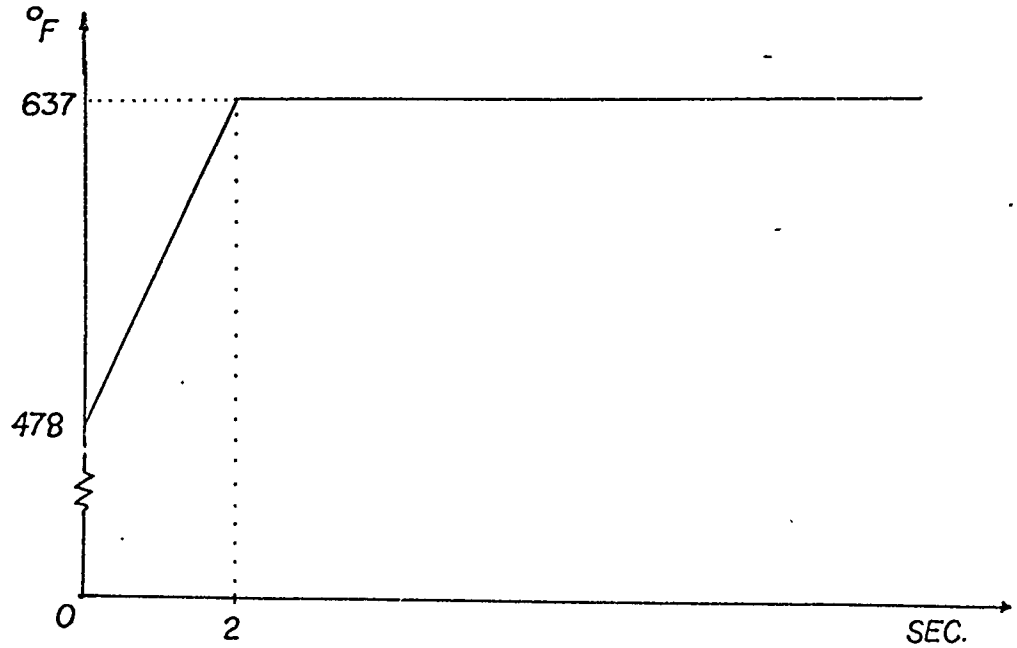
For the structural boundary condition it is assumed that the nodes on the left end cross-section of Fig. 13 are prevented from any axial displacement.

The maximum thermal stresses obtained by the program occur in element 14 of Fig. 13 as follows:

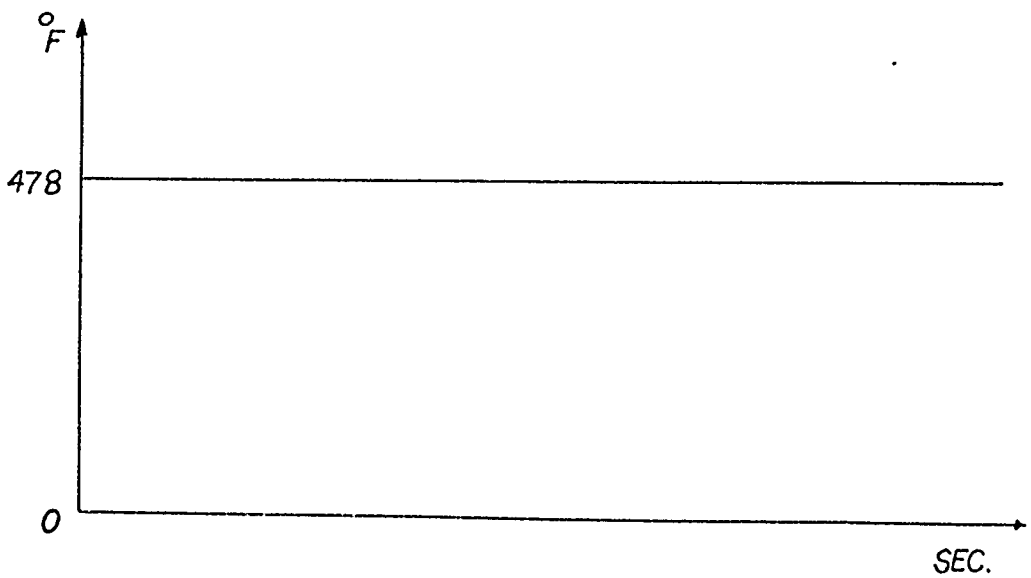
Maximum mean stress = 18.81 ksi. at time = 4 seconds;

Maximum octahedral shearing stress = 11.5 ksi. at time = 18 seconds.

These results appear to be reasonable, but no suitable comparison solution is available.



VARIATION OF ENTRY FLUID TEMPERATURE IN NOZZLE. (SURFACE "1".)

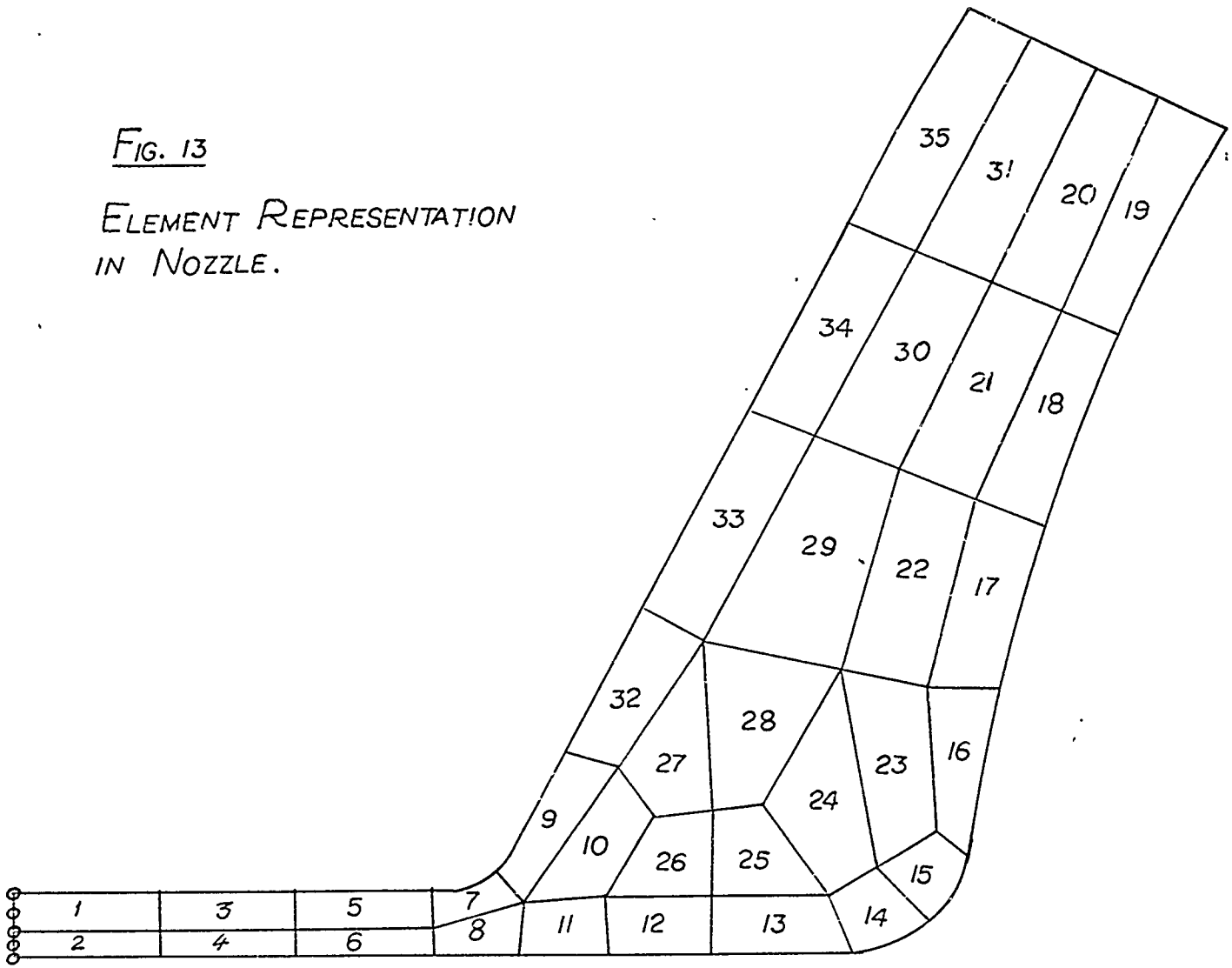


FLUID PAST SURFACE "2".

FIG. 12. Fluid Temperature - Time Histories.

FIG. 13

ELEMENT REPRESENTATION IN NOZZLE.



VII. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

A computer program has been developed for the solution of axisymmetric transient heat conduction and thermal stress problems. The system will accommodate a wide variety of geometric arrangements, thermal and structural boundary conditions, and mechanical loadings.

Although double-precision arithmetic is employed throughout, efficient algorithms for the manipulation and storage of large symmetric banded matrices allow in-core solutions with modest time requirements.

The quadratic isoparametric elements used allow accurate representation of curvilinear boundaries and the stress field. Examples presented show that a small number of elements is generally sufficient to determine stresses with good precision.

B. RECOMMENDATIONS

Incorporation of several additional features would significantly increase the program capabilities. The following extensions are recommended.

1. Material thermal and elastic properties have been assumed constant within each element. Since such properties are generally temperature dependent, provisions should be made for periodically "updating" them during both temperature and stress solutions.

2. The surface heat transfer coefficient, taken as constant in the program, is a function of temperature and flow velocity. Provision should be made to include these effects.

3. The program presently starts every temperature solution with constant initial solid temperature. Provision for an externally specified initial temperature vector should be included.

4. The large quantity of temperature and stress results generated by the program is currently presented as digital printout. Graphical output in the form of two-dimensional contour plots of temperatures and stresses should be provided.

APPENDIX A

APPLICABLE FORMULAS AND EQUATIONS

PART 1

(a): Two dimensional 8-noded (parabolic) isoparametric shape functions.

Corner nodes:

$$N_i = \frac{1}{4}(1 + \xi_o)(1 + \eta_o)(\xi_o + \eta_o - 1)$$

Mid nodes:

$$\xi_i = 0, \quad N_i = \frac{1}{2}(1 - \xi^2)(1 + \eta_o)$$

$$\eta_i = 0, \quad N_i = \frac{1}{2}(1 + \xi_o)(1 - \eta^2)$$

where

$$\xi_o = \xi\,\xi_i\,, \quad \eta_o = \eta\,\eta_i$$

(b): Temperature at a point in terms of the nodal temperatures.

$$T = \sum_{i=1}^{8} N_i T_i$$

(c): Coordinates at a point in terms of the nodal coordinates.

$$R = \sum_{i=1}^{8} N_i R_i$$

$$Z = \sum_{i=1}^{8} N_i Z_i$$

(d): The Jacobian coordinate transformation matrix.

$$\underline{\underline{J}} = \begin{bmatrix} \frac{\partial Z}{\partial \xi} & \frac{\partial R}{\partial \xi} \\ \frac{\partial Z}{\partial \eta} & \frac{\partial R}{\partial \eta} \end{bmatrix}$$

(e): Element of the finite element capacitance matrix.

$$C^e_{ij} = \rho c \int N_i \, N_j \, dV$$

(f): Element of the finite element admittance matrix.

$$y^e_{ij} = k \int \bar{\nabla} N_i \cdot \bar{\nabla} N_j \, dV$$

(g): Elemental volume.

$$dV = 2\pi R \det \underline{\underline{J}} \, d\xi \, d\eta$$

PART 2

(a): One-dimensional 3-noded (parabolic) iso-parametric shape functions.

End nodes $\quad N_i = \frac{1}{2} \xi_o \, (1 + \xi_o)$

Mid node $\quad N_i = (1 - \xi^2)$

where, again, $\xi_o = \xi \, \xi_i$

(b): One-dimensional capacitance and admittance matrices.

$$\underline{\underline{C}}^e = \frac{\rho c \ell}{30} \begin{bmatrix} 4 & 2 & -1 \\ 2 & 16 & 2 \\ -1 & 2 & 4 \end{bmatrix}, \quad \underline{\underline{Y}}^{+e} = \frac{k}{3\ell} \begin{bmatrix} 7 & -8 & 1 \\ -8 & 16 & -8 \\ 1 & -8 & 7 \end{bmatrix} + \begin{bmatrix} h & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

where ℓ is the length of the element.

(c): The element $\underline{v}$ vector for the one-dimensional case.

$$\underline{v}^e = \langle hT_f \; 0 \; 0 \rangle^T$$

PART 3

(a): Strain-displacement relations.

$$\varepsilon_Z = \frac{\partial w}{\partial Z} = \Sigma \frac{\partial N_i}{\partial Z} w_i$$

$$\varepsilon_R = \frac{\partial u}{\partial R} = \Sigma \frac{\partial N_i}{\partial R} u_i$$

$$\varepsilon_\theta = \frac{u}{R} = \frac{\Sigma N_i U_i}{\Sigma N_i R_i}$$

$$\gamma_{RZ} = \frac{\partial u}{\partial Z} + \frac{\partial w}{\partial R} = \Sigma \frac{\partial N_i}{\partial Z} U_i + \Sigma \frac{\partial N_i}{\partial R} w_i$$

where N_i are the same isoparametric shape function as in Part 1, (a).

(b): The $\underline{\underline{B}}$ matrix.

$$\underline{\underline{B}} = \begin{bmatrix} 0 & \frac{\partial N_1}{\partial Z} & 0 & \frac{\partial V_2}{\partial Z} & \cdots\cdots & \frac{\partial N_8}{Z} \\ \frac{\partial N_1}{\partial R} & 0 & \frac{\partial N_2}{\partial Z} & 0 & \cdots\cdots & 0 \\ \frac{N_1}{R} & 0 & \frac{N_2}{R} & 0 & \cdots\cdots & 0 \\ \frac{\partial N_1}{\partial Z} & \frac{\partial N_1}{\partial R} & \frac{\partial N_2}{\partial Z} & \frac{\partial N_2}{\partial R} & \cdots\cdots & \frac{\partial N_8}{\partial R} \end{bmatrix}$$

(c): The element displacement vector.

$$\underline{\delta}^e = \langle u_1 \; w_1 \; u_2 \; w_2 \; \cdots\cdots \; u_8 \; w_8 \rangle^T$$

(d): Elasticity matrix $\underline{\underline{D}}$ for an isotropic material.

$$\underline{\underline{D}} = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1 & \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 0 \\ \frac{\nu}{1-\nu} & 1 & \frac{\nu}{1-\nu} & 0 \\ \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 1 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} \end{bmatrix}$$

APPENDIX B: COMPUTER PROGRAM

```
C
C    A.  IDENTIFICATION
C
C        TITLE                      AXITTS
C                                   (AXISYMMETRIC TRANSIENT
C                                   THERMAL STRESSES.)
C        COMPLETION DATE            JUNE 1972
C        PROGRAMMER                 LT. M. BAKHSHANDEHPOUR
C                                   IMPERIAL IRANIAN  NAVY
C        ADVISER                    DR. ROBERT  E.  NEWTON
C                                   PROF.MECH.ENG.N.P.G.S.
C
C    B.  OBJECTIVE
C
C        THIS PROGRAM IS USED FOR THE FOLLOWING CASES:
C
C        1-  TO EVALUATE TRANSIENT TEMPERATURE IN AN AXI-
C            SYMMETRIC  BODY  FOR  THE COMBINATION OF THE
C            CONSTANT TEMPERATURE, INSULATED AND  CONVEC-
C            TION THERMAL BOUNDARY CONDITION.
C
C        2-  TO EVALUATE THE STRESSES IN AN  AXISYMMETRIC
C            BODY FOR ANY COMBINATION OF AXIAL, PRESSURE,
C            CENTRIFUGAL OR ANY EXTERNALLY SPECIFIED LOAD
C            VECTOR.
C
C        3-  TO EVALUATE TRANSIENT THERMAL STRESSES IN AN
C            AXISYMMETRIC BODY  FOR 'ANY KNOWN' TRANSIENT
C            TEMPERATURE OR USING THE RESULTS OF  STEP 1.
C            THE COMBINATION OF THERMAL LOADING WITH  ANY
C            OTHER LOADING AS IN STEP 2 MAY BE USED ALSO.
C
C    C.  METHOD OF CALCULATION
C
C        THE FINITE ELEMENT METHOD WITH  EIGHT NODED ISO-
C        PARAMETRIC ELEMENTS IS APPLIED FOR THE ANALYSIS.
C
C    D.  USAGE
C
C        REFER TO THE (APPENDIX C) OF THE MECHANICAL ENG.
C        THESIS WRITTEN BY THE PROGRAMMER.
C
```

```
C                                                                       00000510
C                                                                       00000520
C*****************************************************************    00000530
C                                                                       00000540
C                                                                       00000550
C     DESCRIPTION OF THE SYMBOLS.                                       00000560
C                                                                       00000570
C     AJ      IS THE JACOBIAN MATRIX OR ITS INVERSE.                    00000580
C     AL      IS THE COEFFICIENT OF THERMAL EXPANSION , ALPHA.          00000590
C     AXIALF  IS THE AXIAL FORCE.                                       00000600
C     BTE     IS A DUMMY VECTOR FOR CALCULATION OF TEMPERATURE.         00000610
C     BDRYC   IS THE OUTSIDE THERMAL BOUNDARY CONDITION                 00000620
C     BDRYI   IS THE INSIDE  THERMAL BOUNDARY CONDITION.                00000630
C     C       IS THE THERMAL CAPACITANCE MATRIX OR C+.5*DTI*Y           00000640
C     CE      IS THE ELEMENT CAPACITANCE MATRIX.                        00000650
C     COPY    IS A COPY OF PART OF STIFFNESS MATRIX.                    00000660
C     D       IS THE STANDARD D MATRIX.                                 00000670
C     DL      IS THE LOWER TRIANGLE OF D MATRIX.                        00000680
C     DNZR    IS AN ARRAY OF THE DERIVATIVES OF SHAPE FUNCTIONS         00000690
C             RESPECT TO Z AND R COORDINATES.                           00000700
C     DETA    IS THE ARRAY OF THE DERIVATIVES OF SHAPE FUNCTIONS        00000710
C             WITH RESPECT TO ETA.                                      00000720
C     DTI     IS THE TIME INCREMENT USED IN INTEGRATION.                00000730
C     DXI     IS THE ARRAY OF THE DERIVATIVES OF THE SHAPE              00000740
C             FUNCTIONS WITH RESPECT TO XI.                             00000750
C     DENS    IS THE DENSITY.                                           00000760
C     E       IS THE YOUNG'S MODULUS OF ELASTICITY.                     00000770
C     EPSO    IS THE LOCAL INITIAL STRAIN VECTOR.                       00000780
C     ESM     IS THE ELEMENT STIFFNESS MATRIX.                          00000790
C     F       IS THE ARRAY OF VARIOUS LOAD VECTORS.                     00000800
C     FE      IS THE ELEMENT LOAD VECTOR.                               00000810
C     GC      IS THE GAUSS COEFFICIENT.                                 00000820
C     GX      IS THE GAUSS ABSCISSA.                                    00000830
C     HTCI    IS INSIDE  HEAT TRANSFER COEFFICIENT                      00000840
C     HTCO    IS OUTSIDE HEAT TRANSFER COEFFICIENT                      00000850
C     IVEC    IS THE NUMBER OF STORED TEMPERATURE VECTORS.               00000860
C     INTP    IS THE INTERVAL OF TEMPERATURE PRINT                      00000870
C     JVEC    IS THE LOCATION INDICATION OF VARIOUS LOAD                00000880
C             VECTORS WITHIN F .                                        00000890
C     NCFIT   IS THE TOTAL NUMBER OF NODES IN CONTACT WITH              00000900
C             INSIDE FLUID.                                             00000910
C     NCFOT   IS THE TOTAL NUMBER OF NODES IN CONTACT WITH              00000920
C             OUT SIDE FLUID.                                           00000930
C     NET     IS THE NUMBER OF ELEMENTS.                                00000940
C     NDF     IS THE NUMBER OF DEGREES OF FREEDOM.                      00000950
C     NGP     IS THE NUMBER OF GAUSS POINTS FOR INTEGRATION.            00000960
C     NNE     IS THE NUMBER OF NODES PER ELEMENT                        00000970
C     NNLT    IS THE TOTAL NUMBER OF NODES LONGITUDINALLY FIXED.        00000980
```

```
C        NNRT    IS THE TOTAL NUMBER OF NODES RADIALLY FIXED                    00000990
C        NPROB   IS THE NUMBER OF PROBLEMS                                      00001000
C        NPASS   IS THE NUMBER OF STEPS OF TIME INTEGRATION.                    00001010
C        NPNT    IS THE TOTAL NUMBER OF PRESSURE NODES.                         00001020
C        NCFI    IS THE ARRAY OF NODES IN CONTACT WITH FLUID INSIDE             00001030
C        NCFO    IS THE ARRAY OF NODES IN CONTACT WITH FLUID OUTSIDE            00001040
C        OMEGA   IS THE NUMBER OF REVOLUTIONS PER MINUTE.                       00001050
C        POI     IS THE POISSON'S RATIO.                                        00001060
C        PRES    IS THE PRESSURE.                                               00001070
C        R       IS THE ARRAY OF THE RADIAL COORDINATES.                        00001080
C        SHP     IS THE ARRAY OF THE SHAPE FUNCTIONS.                           00001090
C        SHT     IS THE SPECIFIC HEAT.                                          00001100
C        SSM     IS THE SYSTEM STIFFNESS MATRIX.                                00001110
C        STOR    IS THE ARRAY OF STORED TEMPERATURE VECTORS.                    00001120
C        SMAX    IS THE MAX. MEAN STRESS.                                       00001130
C        SHMAX   IS THE MAX. OCTAHEDRAL SHEAR STRESS.                           00001140
C        TINIT   IS THE INITIAL TEMPERATURE OF THE BODY.                        00001150
C        TK      IS THE THERMAL CONDUCTIVITY.                                   00001160
C        TMAXM   IS THE TIME WHEN MAX. MEAN STRESS OCCURS.                      00001170
C        TMAXO   IS THE TIME WHEN MAX. OCTAHEDRAL SHEAR                         00001180
C                STRESS OCCURS.                                                 00001190
C        TREF    IS THE REFERENCE TEMPERATURE.                                  00001200
C        Z       IS THE ARRAY OF THE LONGITUDINAL COORDINATES.                  00001210
C        ***********************************************************            00001220
C                                                                               00001230
      IMPLICIT  REAL * 8 (A-H,O-Z)                                              00001240
      REAL   * 4   ENTEO,ENTEI,ENTIO,ENTII,PLOT                                 00001250
      COMMON /ONE/ AJ(2,2),B(4,16),COPY(37,298),D(4,4),DL(4,4),                 00001260
     1DNZR(2,8),ESM(16,16),F(24,298),SSM(298,66),STOR(20,150)                   00001270
C                                                                               00001280
      COMMON /TWO/ AL(5),DENS(5),DETA(8),DXI(8),E(5),EPSO(4),                   00001290
     1FE(16),GC(4),GX(4),POI(5),R(149),SHP(8),SHT(5),TK(5),Z(149)               00001300
C                                                                               00001310
      COMMON /TRE/ AXIALF,DTI,EVERY,FORS1,OMEGA,HTCI,HTCO,                      00001320
     1POSI,PRES,Q1,Q2,Q3,SHMAX,SMAX,TIME,TIME1,TINIT,TMAXM,TREF,                00001330
     2TMAXO,RMAXM,RMAXO,ZMAXM,ZMAXO,Q4,BIG                                      00001340
C                                                                               00001350
      COMMON /FOR/ NN(40,9)                                                     00001360
C                                                                               00001370
      COMMON /FIV/ NCFI(37),NCFO(37),NNR(37),NRE(37),NNL(37),                   00001380
     1NPN(37)                                                                   00001390
      COMMON /SIX/ IBAND,IEXT,INTP,IVEC,JVEC,KK,LOAD,NCFIT,NCFOT,               00001400
     1NCHECK,NET,NDF,NGP,NNE,NNLT,NNRT,NPASS,NPROB,NRPPI,NRPPO,                 00001410
     3NNT,NNRE,NPNT,IPLANE,NBAND                                                00001420
C                                                                               00001430
      EQUIVALENCE  (C,SSM) , (Y,SSM(1,18)) , (T,SSM(1,35)) ,                    00001440
                                                                                00001450
                                                                                00001460
```

```
     1(BTE,SSM(1,36)) , (TEMP,SSM(1,37)) , (V,SSM(1,38)) ,                    00001470
     2(FT1,SSM(1,39)) , (FT2,SSM(1,40)) , (CE,SSM(1,41)) ,                    00001480
     3(YE,SSM(1,42)) , (OFF1,SSM(1,43)) , (OFF2,SSM(1,44)) ,                  00001490
     4(DIAG,SSM(1,45)) , (BDRYI,SSM(1,46)) , (BDRYO,SSM(1,47)) ,              00001500
     5(ENTII,SSM(1,48)) , (ENTIO,SSM(1,49)), (ENTEI,SSM(1,50)),               00001510
     6(ENTEO,SSM(1,51)),(VOL,SSM(1,52)),(VOLO,SSM(1,53))                      00001520
C                                                                             00001530
      DIMENSION  C(149,33),Y(149,33),T(149),BTE(149),TEMP(149),               00001540
     1V(149),FT1(90),FT2(90),CE(8,8),YE(8,8),OFF1(37),OFF2(37),               00001550
     2DIAG(37),BDRYI(16,5),BDRYO(16,5),VOL(36),VOLO(36),ENTEO(75),            00001560
     3ENTEI(75),ENTIO(75),ENTII(75)                                           00001570
C                                                                             00001580
C                                                                             00001590
      CALL   INPUT                                                            00001600
C                                                                             00001610
C        TEMPERATURE PROBLEM                                                  00001620
C                                                                             00001630
C                                                                             00001640
      DO  2  L = 1 , NPROB                                                    00001650
C                                                                             00001660
C                                                                             00001670
      DO  5  I = 1 , 75                                                       00001680
      ENTEO(I) = 0.0                                                          00001690
      ENTEI(I) = 0.0                                                          00001700
      ENTIO(I) = 0.0                                                          00001710
    5 ENTII(I) = 0.0                                                          00001720
C                                                                             00001730
C                                                                             00001740
      WRITE  (6,99)  L                                                        00001750
   99 FORMAT  (1H1,//,50X,'PROBLEM NUMBER ',I5,/,50X,                         00001760
     120('*'),/,50X,20('*'),//)                                               00001770
      CALL   PROB                                                             00001780
C                                                                             00001790
      IF  ((Q4.LT.0.D0).OR.(IEXT.NE.0))  GO  TO  3                            00001800
C                                                                             00001810
      CALL CANDY                                                              00001820
      DO  1  K = 1 , NPASS                                                    00001830
      CALL  FLOW                                                              00001840
C                                                                             00001850
      CALL  FORMV                                                             00001860
C                                                                             00001870
      CALL  TEMPER                                                            00001880
    1 CONTINUE                                                                00001890
C                                                                             00001900
      IF  (Q2.GT.0.D0)  CALL  PLOT(ENTIO,ENTEO,0)                             00001910
      IF  (Q3.GT.0.D0)  CALL  PLOT(ENTII,ENTEI,1)                             00001920
C                                                                             00001930
      IF  (Q4.GT.0.D0)  GO  TO  2                                             00001940
```

```
C                                                                       00001950
C         STRESS PROBLEM                                                00001960
C                                                                       00001970
C                                                                       00001980
    3 CALL   STIFF                                                      00001990
      CALL   FORMF                                                      00002000
      IF  (OMEGA.NE.0.D0) CALL  CENTF                                   00002010
      IF  (PRES.NE.0.D0)  CALL  PRESS                                   00002020
      CALL   DISPL                                                      00002030
      CALL   STRESS                                                     00002040
C                                                                       00002050
C                                                                       00002060
      WRITE  (6,100)                                                    00002070
  100 FORMAT (1H1,///,2(25X,54('*'),/),2(25X,'**',50X,'**',/),          00002080
     125X,'**',50X,'**')                                                00002090
      WRITE  (6,101) TMAXM , SMAX , RMAXM , ZMAXM                       00002100
  101 FORMAT (25X,'**    AT TIME = ',F10.2,27X,'**',/,25X,              00002110
     1'**    MAX. MEAN STRESS IS ',1PD14.6,13X,'**',/,25X,              00002120
     2'**    AND IT OCCURS AT THE POINT WHERE',15X,'**',/,              00002130
     325X,'**    R = ',1PD14.6,8X,'Z = ',1PD14.6,3X,'**',/,             00002140
     48(25X,'**',50X,'**',/),25X,'**',50X,'**')                         00002150
      WRITE  (6,102) TMAXO , SHMAX , RMAXO , ZMAXO                      00002160
  102 FORMAT (25X,'**    AT TIME = ',F10.2,27X,'**',/,25X,              00002170
     1'**    MAX. OCTAHEDRAL SHEAR STRESS IS',1PD14.6,2X,               00002180
     2'**',/,25X,'**    AND IT OCCURS AT THE POINT WHERE',              00002190
     315X,'**',/,25X,'**    R= ',1PD14.6,8X,'Z = ',1PD14.6,             00002200
     44X,'**',/,4(25X,'**',50X,'**',/),2(25X,54('*'),/))                00002210
C                                                                       00002220
C                                                                       00002230
C                                                                       00002240
    2 CONTINUE                                                          00002250
      STOP                                                              00002260
      END                                                               00002270
C                                                                       00002280
```

```
      SUBROUTINE  INPUT
C
C     ************************************************************
C     *                                                          *
C     THIS SUBROUTINE READS AND WRITES THE INPUT DATA OF         *
C     THE GEOMETRY AND MATERIAL PROPERTIES.IT CALCULATES         *
C     THE MID NODE COORDINATES BASED ON STRAIGHT AND THE         *
C     SYSTEM OF UNITS IS SELECTED HERE.                          *
C     AT THE END THE BANDWIDTH OF THE STIFFNESS  MATRIX          *
C     IS DETERMINED.                                             *
C                                                                *
C     ************************************************************
C
C
C
      IMPLICIT  REAL  * 8 (A-H,O-Z)
      REAL  *  4  HCO,VCO,PLOTZR
C
      COMMON /ONE/ AJ(2,2),B(4,16),COPY(37,298),D(4,4),DL(4,4),
     1DNZR(2,8),ESM(16,16),F(24,298),SSM(298,66),STOR(20,150)
C
      COMMON /TWO/ AL(5),DENS(5),DETA(8),DXI(8),E(5),EPSO(4),
     1FE(16),GC(4),GX(4),POI(5),R(149),SHP(8),SHT(5),TK(5),Z(149)
C
      COMMON /TRE/ AXIALF,DTI,EVERY,FORS1,OMEGA,HTCI,HTCO,
     1POSI,PRES,Q1,Q2,Q3,SHMAX,SMAX,TIME,TIME1,TINIT,TMAXM,TREF,
     2TMAXO,RMAXM,RMAXO,ZMAXM,ZMAXO,Q4,BIG
C
      COMMON /FOR/ NN(40,9)
C
      COMMON /FIV/ NCFI(37),NCFO(37),NNR(37),NRE(37),NNL(37),
     1NPN(37)
      COMMON /SIX/ IBAND,IEXT,INTP,IVEC,JVEC,KK,LOAD,NCFIT,NCFOT,
     1NCHECK,NET,NDF,NGP,NNE,NNLT,NNRT,NPASS,NPROB,NRPPI,NRPPO,
     3NNT,NNRE,NPNT,IPLANE,NBAND
C
      EQUIVALENCE  (C,SSM) , (Y,SSM(1,18)) , (T,SSM(1,35)) ,
     1(BTE,SSM(1,36)) , (TEMP,SSM(1,37)) , (V,SSM(1,38)) ,
     2(FT1,SSM(1,39)) , (FT2,SSM(1,40)) , (CE,SSM(1,41)) ,
     3(YE,SSM(1,42)) , (OFF1,SSM(1,43)) , (OFF2,SSM(1,44)) ,
     4(DIAG,SSM(1,45)) , (BDRYI,SSM(1,46)) , (BDRYO,SSM(1,47)) ,
     5(ENTII,SSM(1,48)) , (ENTIO,SSM(1,49)), (ENTEI,SSM(1,50)),
     6(ENTEO,SSM(1,51)),(VOL,SSM(1,52)),(VOLO,SSM(1,53))
      EQUIVALENCE  (HCO,SSM(1,54)) , (VCO,SSM(1,55))
C
      DIMENSION  C(149,33),Y(149,33),T(149),BTE(149),TEMP(149),
```

```
     1V(149),FT1(90),FT2(90),CE(8,8),YE(8,8),OFF1(37),OFF2(37),                00002750
     2DIAG(37),BDRYI(16,5),BDRYO(16,5),VOL(36),VOLO(36),ENTEO(75),             00002760
     3ENTEI(75),ENTIO(75),ENTII(75)                                            00002770
      DIMENSION  HCO(149) , VCO(149)                                           00002780
C                                                                              00002790
      DATA  BRIT/'BRIT'/                                                       00002800
C                                                                              00002810
      READ (5,301) NET,NNT,NCN,NOFN,NMAT,NPROB                                 00002820
  301 FORMAT  (8I5)                                                            00002830
      NNE = 8                                                                  00002840
C                                                                              00002850
      DO  4  I = 1 , NCN                                                       00002860
    4 READ  (5,300)  M,R(M),Z(M)                                               00002870
  300 FORMAT  (I5,2F10.5)                                                      00002880
      K=NNE + 1                                                                00002890
      READ (5,302)    ((NN(I,J),J=1,K),I=1,NET)                                00002900
  302 FORMAT (9I5)                                                             00002910
C                                                                              00002920
CC                                                                             00002930
CC    CALCULATION OF MID-NODE COORDINATES.                                     00002940
CC                                                                             00002950
C                                                                              00002960
      DO  5  L = 1 , NET                                                       00002970
      R(NN(L,2)) = (R(NN(L,1))+R(NN(L,3))) / 2.D0                              00002980
      Z(NN(L,2)) = (Z(NN(L,1))+Z(NN(L,3))) / 2.D0                              00002990
      R(NN(L,4)) = (R(NN(L,3))+R(NN(L,5))) / 2.D0                              00003000
      Z(NN(L,4)) = (Z(NN(L,3))+Z(NN(L,5))) / 2.D0                              00003010
      R(NN(L,6)) = (R(NN(L,5))+R(NN(L,7))) / 2.D0                              00003020
      Z(NN(L,6)) = (Z(NN(L,5))+Z(NN(L,7))) / 2.D0                              00003030
      R(NN(L,8)) = (R(NN(L,7))+R(NN(L,1))) / 2.D0                              00003040
    5 Z(NN(L,8)) = (Z(NN(L,7))+Z(NN(L,1))) / 2.D0                              00003050
C                                                                              00003060
C                                                                              00003070
      IF  (NOFN. EQ . 0)  GO  TO  7                                            00003080
      DO  6  I = 1 , NOFN                                                      00003090
    6 READ  (5,300)  M,R(M),Z(M)                                               00003100
C                                                                              00003110
    7 CONTINUE                                                                 00003120
C                                                                              00003130
      DO  19  I = 1 , NNT                                                      00003140
      HCO(I) = Z(I)                                                            00003150
      VCO(I) = R(I)                                                            00003160
   19 CONTINUE                                                                 00003170
C                                                                              00003180
C                                                                              00003190
      CALL  PLOTZR(HCO,VCO,NNT)                                                00003200
C                                                                              00003210
C                                                                              00003220
```

```
      DO  1  I = 1 , NMAT                                                00003230
    1 READ  (5,303) E(I),AL(I),POI(I),TK(I),DENS(I),SHT(I)               00003240
  303 FORMAT  (2D12.4,4F8.3)                                             00003250
C                                                                        00003260
CC                                                                       00003270
CC    ARE THE UNITS IN METRIC OR BRITISH UNIT SYSTEM?                    00003280
C                                                                        00003290
      READ (5,310) Q1                                                    00003300
  310 FORMAT  (A4)                                                       00003310
C                                                                        00003320
      WRITE  (6,200)                                                     00003330
  200 FORMAT (1H1,45X,'** TRANSIENT THERMAL STRESS PROBLEM',             00003340
     1' **',/,49X,32('_'),//)                                            00003350
C                                                                        00003360
CC                                                                       00003370
      WRITE  (6,201)  NET,NNT,NCN,NMAT,NPROB                             00003380
  201 FORMAT  (/,15X,'TOTAL NUMBER OF ELEMENTS.....=',I5,                00003390
     1/,15X,'TOTAL NUMBER OF NODES.........=',I5,/,15X,                  00003400
     2'TOTAL NUMBER OF CORNER NODES.=',I5,/,15X,'TOTAL ',                00003410
     3'NUMBER OF MATERIALS....=',I5,/,15X,'TOTAL NUMBER ',               00003420
     4'OF PROBLEMS.....=',I5,//)                                         00003430
C                                                                        00003440
CC                                                                       00003450
      WRITE  (6,204)                                                     00003460
  204 FORMAT (//,15X,'ELEMENT NO.',8X,'GLOBAL NODE NUMBER',              00003470
     132X,'MATERIAL NO.',/)                                              00003480
C                                                                        00003490
CC                                                                       00003500
      DO  2  I=1 , NET                                                   00003510
    2 WRITE  (6,205)  I,(NN(I,J),J=1,K)                                  00003520
  205 FORMAT (15X,I5,10X,9(I5,2X))                                       00003530
C                                                                        00003540
CC                                                                       00003550
      IF  (Q1.NE.BRIT)  GO  TO  10                                       00003560
C                                                                        00003570
      WRITE (6,211)                                                      00003580
      WRITE  (6,213)                                                     00003590
      GO TO 11                                                           00003600
C                                                                        00003610
CC                                                                       00003620
   10 WRITE  (6,212)                                                     00003630
      WRITE  (6,215)                                                     00003640
   11 WRITE (6,202)                                                      00003650
C                                                                        00003660
CC                                                                       00003670
  202 FORMAT(/,15X,'THE RADIAL AND LONGITUDINAL',                        00003680
     1' COORDINATES',///,15X,'NODE NO.',14X,'**R**',                     00003690
     212X,'**Z**',/)                                                     00003700
```

```
C                                                                       00003710
C                                                                       00003720
      WRITE  (6,216) (I,R(I),Z(I),I=1,NNT)                              00003730
  216 FORMAT  (16X,I4,13X,F10.4,6X,F10.4)                               00003740
C                                                                       00003750
  211 FORMAT (//,15X,'** THE FOLLOWING UNITS ARE USED **',/,            00003760
     115X,'NAME',16X,'SYMBOL',14X,'UNIT',/,15X,4('*'),16X,              00003770
     2'******',14X,'****',///,15X,'TIME',16X,'TIME,DII',                00003780
     312X,'SECOND',/,15X,'TEMPERATURE',9X,'T,TEMP,FT',11X,              00003790
     4'DEGREE FAHRENHEIT',/,15X,'VELOCITY',12X,'VEL',17X,               00003800
     5'FEET PER SECOND',/,15X,'COORDINATES',9X,'R,Z',                   00003810
     617X,'INCHES',/,15X,'MOD. OF ELASTICITY',2X,'E',19X,               00003820
     7'POUND FORCE/SQ.INCHES',/,15X,'POISSON S RATIO',5X,               00003830
     8'POI',17X,'DIMENSIONLESS')                                        00003840
C                                                                       00003850
C                                                                       00003860
  213 FORMAT  (15X,'COEF.OF THER.EXP.',                                 00003870
     13X,'AL',18X,'1/DEGREE F.',/,15X,'THER.CONDUCTIVITY',              00003880
     23X,'TK',18X,'BTU/FT.HR.,DEG.F',/,15X,'DENSITY',13X,               00003890
     3'DENS',16X,'LB/FT.CUBE',/,15X,'SPECIFIC HEAT',7X,                 00003900
     4'SHT',17X,'BTU/LB.DEG.F',/,15X,'HEAT TRANSFER COEF',              00003910
     52X,'HTCO,HTCI',11X,'BTU/FT.SQ,HR.,DEG.F',/)                       00003920
C                                                                       00003930
C                                                                       00003940
  212 FORMAT (//,15X,'THE FOLLOWING UNITS ARE BEING USED',              00003950
     1//,15X,'NAME',16X,'SYMBOL',14X,'UNIT',/,15X,'****',               00003960
     216X,'******',14X,'****',///,15X,'TIME',16X,'TIME,DTI',            00003970
     312X,'SECOND',/,15X,'TEMPERATURE',9X,'T,TEMP,FT',11X,              00003980
     4'CELSIUS   ',/,15X,'VELOCITY',12X,'VEL',17X,'METER/',             00003990
     5'SEC',/,15X,'COORDINATES',9X,'R,Z',17X,'CENTIMETER',              00004000
     6/,15X,'MOD.OF ELASTICITY',3X,'E',19X,'KG./CM.SQ',/,               00004010
     715X,'COEF.OF THER.EXP.',3X,'AL',18X,'1/CELSIUS')                  00004020
C                                                                       00004030
C                                                                       00004040
  215 FORMAT  (15X,                                                     00004050
     1    'THER.CONDUCTIVITY',3X,'TK',18X,'CAL/HR.,M.,CEL',            00004060
     2'SIUS   ',/,15X,'DENSITY',13X,'DENS',16X,'GM./CM.C',              00004070
     3'UBE',/,15X,'SPECIFIC HEAT',7X,'SHT',17X,'CAL./KG.',              00004080
     4'CELSIUS   ',/,15X,'HEAT TRANSFER COEF',2X,'HTCO,H',              00004090
     5'TCI',11X,'CAL/HR,M.SQ,CELSIUS',/)                                00004100
C                                                                       00004110
      WRITE  (6,206)                                                    00004120
  206 FORMAT (//,15X,'* PROPERTIES OF THE MATERIALS *',//,              00004130
     115X,'MATERIAL',5X,'MODULUS OF',5X,'POISSON S',5X,                 00004140
     2'THERM.EXP,',5X,'THERMAL',10X,'VOLUME',6X,'SPECIFIC')             00004150
C                                                                       00004160
C                                                                       00004170
      WRITE  (6,207)                                                    00004180
```

```
  207 FORMAT (15X,'NUMBER',7X,'ELASTICITY',5X,'RATIO',9X,                  00004190
     1'COEFFICIENT',4X,'CONDUCTIVITY',5X,'DENSITY',5X,                     00004200
     2'HEAT',//)                                                           00004210
C                                                                          00004220
C                                                                          00004230
      DO  3  I=1,NMAT                                                      00004240
    3 WRITE  (6,209) I,E(I),POI(I),AL(I),TK(I),DENS(I),SHT(I)              00004250
  209 FORMAT (14X,I5,6X,F10.1,6X,F9.5,5X,F10.7,6X,F8.4,9X,                 00004260
     1F8.4,5X,F8.4,/)                                                      00004270
C                                                                          00004280
C                                                                          00004290
C        CONVERSION OF THE SYSTEM OF UNITS.                                00004300
C                                                                          00004310
      IF  (Q1.EQ.BRIT)  GO  TO  60                                         00004320
      TREF = 20.DO                                                         00004330
      DO  61  I = 1 , NMAT                                                 00004340
      DENS(I) = DENS(I) / 1000.DO                                          00004350
   61 E(I) = E(I) * 100.DO                                                 00004360
      GO  TO  63                                                           00004370
   60 TREF = 68.DO                                                         00004380
      DO  62  I = 1 , NMAT                                                 00004390
      DENS(I) = DENS(I) / 1728.DO                                          00004400
   62 TK(I) = TK(I) / 14200.DO                                             00004410
   63 CONTINUE                                                             00004420
C                                                                          00004430
C                                                                          00004440
C          DETERMINATION  OF  THE  HALF-BAND-WIDTH  OF THE                 00004450
C          SYSTEM  STIFFNESS MATRIX                                        00004460
C                                                                          00004470
C                                                                          00004480
      IDIF = 0                                                             00004490
      NNE1 = NNE - 1                                                       00004500
      DO  22  I = 1 , NET                                                  00004510
      IDIF1 = 0                                                            00004520
C                                                                          00004530
C                                                                          00004540
      DO  21  J = 2 , NNE1                                                 00004550
      IDIF2 = NN(I,1) - NN(I,J)                                            00004560
      IF  (IDIF2. LT . 0 )  IDIF2 = -IDIF2                                 00004570
      IF  (IDIF2.GT.IDIF1)  IDIF1 = IDIF2                                  00004580
   21 CONTINUE                                                             00004590
C                                                                          00004600
      IF  (IDIF1.GT.IDIF)   IDIF = IDIF1                                   00004610
C                                                                          00004620
   22 CONTINUE                                                             00004630
C                                                                          00004640
      NBAND = IDIF + 1                                                     00004650
      IBAND = 2 * NBAND                                                    00004660
```

```
      NDF = 2 * NNT                                                     00004670
C                                                                       00004680
      WRITE  (6,220)  IBAND , NDF                                       00004690
  220 FORMAT (/,15X,'HALF-BAND-WIDTH OF THE STIFFNESS MATRIX IS',       00004700
     1I5,/,15X,'NUMBER OF DEGREES OF FREEDUM OF THE SYSTEM IS',I5,/)    00004710
C                                                                       00004720
      RETURN                                                            00004730
      END                                                               00004740
C                                                                       00004750
C                                                                       00004760
```

```
      SUBROUTINE  PROB                                                          00004770
C                                                                               00004780
CCCCCCCCCCCCCCCC                                                                00004790
C     ************************************************************            00004800
C     *                                                          *            00004810
C     *  FOR EACH PROBLEM , IN THIS SUBROUTINE THE THERMAL       *            00004820
C     *  INITIAL AND BOUNDARY CONDITIONS AND  THE  VARIOUS       *            00004830
C     *  OTHER TYPES OF LOADING WITH SPECIFIED  STRUCTURAL       *            00004840
C     *  BOUNDARY CONDITIONS ARE BEING READ IN AND PRINTED       *            00004850
C     *  OUT. THE STEP SIZE OF TIME INTEGRATION , INTERVAL       *            00004860
C     *  OF TEMPERATURE PRINTS IF REQUIRED , AND THE NUMBER      *            00004870
C     *  OF THERMAL LOAD VECTORS WANTED ARE READ IN HERE.        *            00004880
C     *                                                          *            00004890
C     ************************************************************            00004900
C                                                                               00004910
      IMPLICIT  REAL * 8 (A-H,O-Z)                                              00004920
C                                                                               00004930
      COMMON /ONE/ AJ(2,2),B(4,16),COPY(37,298),D(4,4),DL(4,4),                 00004940
     1DNZR(2,8),ESM(16,16),F(24,298),SSM(298,66),STOR(20,150)                   00004950
C                                                                               00004960
      COMMON /TWO/ AL(5),DENS(5),DETA(8),DXI(8),E(5),EPSO(4),                   00004970
     1FE(16),GC(4),GX(4),PCI(5),R(149),SHP(8),SHT(5),TK(5),Z(149)               00004980
C                                                                               00004990
      COMMON /TRE/ AXIALF,DTI,EVERY,FORS1,OMEGA,HTCI,HTCO,                      00005000
     1POSI,PRES,Q1,Q2,Q3,SHMAX,SMAX,TIME,TIME1,TINIT,TMAXM,TREF,                00005010
     2TMAXO,RMAXM,RMAXO,ZMAXM,ZMAXO,Q4,BIG                                      00005020
C                                                                               00005030
      COMMON /FOR/ NN(40,9)                                                     00005040
C                                                                               00005050
      COMMON /FIV/ NCFI(37),NCFO(37),NNR(37),NRE(37),NNL(37),                   00005060
     1NPN(37)                                                                   00005070
      COMMON /SIX/ IBAND,IEXT,INTP,IVEC,JVEC,KK,LOAD,NCFIT,NCFOT,               00005080
     1NCHECK,NET,NDF,NGP,NNE,NNLT,NNRT,NPASS,NPROB,NRPPI,NRPPO,                 00005090
     3NNT,NNRE,NPNT,IPLANE,NBAND                                                00005100
C                                                                               00005110
      EQUIVALENCE  (C,SSM) , (Y,SSM(1,18)) , (T,SSM(1,35)) ,                    00005120
     1(BTE,SSM(1,36)) , (TEMP,SSM(1,37)) , (V,SSM(1,38)) ,                      00005130
     2(FT1,SSM(1,39)) , (FT2,SSM(1,40)) , (CE,SSM(1,41)) ,                      00005140
     3(YE,SSM(1,42)) , (OFF1,SSM(1,43)) , (OFF2,SSM(1,44)) ,                    00005150
     4(DIAG,SSM(1,45)) , (BDRYI,SSM(1,46)) , (BDRYO,SSM(1,47)) ,                00005160
     5(ENTII,SSM(1,48)) , (ENTIO,SSM(1,49)), (ENTEI,SSM(1,50)),                 00005170
     6(ENTEO,SSM(1,51)),(VOL,SSM(1,52)),(VOLO,SSM(1,53))                        00005180
C                                                                               00005190
      DIMENSION  C(149,33),Y(149,33),T(149),BTE(149),TEMP(149),                 00005200
     1V(149),FT1(90),FT2(90),CE(8,8),YE(8,8),OFF1(37),OFF2(37),                 00005210
                                                                               00005220
```

```
     2DIAG(37),BDRYI(16,5),BDRYO(16,5),VOL(36),VOLO(36),ENTEO(75),        00005230
     3ENTEI(75),ENTIO(75),ENTII(75)                                       00005240
C                                                                         00005250
      DATA  BRIT/'BRIT'/                                                  00005260
C                                                                         00005270
      TIME = 0.D0                                                         00005280
      JVEC = 0                                                            00005290
      SMAX = 0.D0                                                         00005300
      SHMAX = 0.D0                                                        00005310
C                                                                         00005320
      NGP = 4                                                             00005330
      GX(1) = -.861136311594053                                           00005340
      GX(2) = -GX(1)                                                      00005350
      GX(3) = .339981043584856                                            00005360
      GX(4) = -GX(3)                                                      00005370
      GC(1) = .347854845137454                                            00005380
      GC(2) = GC(1)                                                       00005390
      GC(3) = .652145154862546                                            00005400
      GC(4) = GC(3)                                                       00005410
      PAI = 3.141592653589793                                             00005420
C                                                                         00005430
CCC                                                                       00005440
CCC      NNRE IS THE TOTAL NUMBER OF NODES AT RIGHT HAND END.             00005450
CCC      NRE(I) IS THE ARRAY OF GLOBAL NODE NUMBER FOR THE                00005460
CCC      RIGHT HAND END.                                                  00005470
CCC      IPLANE.GT.0  IF PLANES REMAIN PLANE AT THE ENDS.                 00005480
CCC      IPLANE.EQ.0  IF ONE END IS FREE OR RADIALLY FIXED.               00005490
CCC                                                                       00005500
CCC      IEXT  IS AN INDICATION IF TEMPERATURE VECTORS ARE                00005510
CCC      GENERATED OR BEING READ IN AS FOLLOWES.                          00005520
CCC      IEXT.EQ.0   TEMPERATURE VECTORS ARE GENERATED.                   00005530
CCC      IEXT.GT.0   TEMPERATURE VECTORS ARE INPUTED.                     00005540
CCC                                                                       00005550
CC                                                                        00005560
C                                                                         00005570
      READ (5,101)OMEGA,PRES,IEXT,LOAD,NNLT,NNRT,IPLANE                  00005580
C                                                                         00005590
      IF  (OMEGA.EQ.0.D0)  GO  TO  2                                      00005600
      JVEC = JVEC + 1                                                     00005610
      WRITE  (6,203)  OMEGA                                               00005620
    2 N = 0                                                               00005630
      IF  (PRES.EQ.0.D0)  GO  TO  6                                       00005640
      READ  (5,102)  NPNT,(NPN(I),I=1,NPNT)                               00005650
C                                                                         00005660
CC       NPNT  IS THE TOTAL NUMBER OF PRESSURE NODES                      00005670
CC       NPN(I)  IS THE GLOBAL NODE NUMBER OF THE PRESSURE NODE I .       00005680
C                                                                         00005690
      WRITE  (6,444)  NPNT,(NPN(I),I=1,NPNT)                              00005700
```

```
  444 FORMAT  (15X,'TOTAL NUMBER OF PRESSURE NODES IS ',I5,' ',          00005710
     1'AND THEY ARE AS :',/,3(15X,20(I3,' ,'),/),/)                     00005720
      JVEC = JVEC + 1                                                   00005730
      WRITE  (6,207)  PRES                                              00005740
    6 CONTINUE                                                          00005750
C                                                                       00005760
      IF  (LOAD.EQ.0)  GO  TO  50                                       00005770
      JVEC = JVEC + 1                                                   00005780
      L = IVEC + JVEC                                                   00005790
      READ  (5,108)  (F(L,I) , I=1 , NDF)                               00005800
      WRITE  (6,333) (F(L,I) , I=1 , NDF)                               00005810
  333 FORMAT  (/,15X,'THE EXTERNAL LOAD VECTOR IN ORDER OF ',           00005820
     1'THE NODE NUMBERS IS :',/,5(15X,F10.4),/)                         00005830
   50 CONTINUE                                                          00005840
C                                                                       00005850
      IF (NNLT.NE.0) READ (5,304) (NNL(I),I=1,NNLT)                     00005860
      IF (NNLT.NE.0) WRITE (6,308) NNLT,(NNL(I),I=1,NNLT)               00005870
  308 FORMAT (/,15X,'TOTAL NUMBER OF NODES FIXED IN THE ',              00005880
     1'LONGITUDINAL DIRECTION IS ',I5,' AND THEY ARE AS:',              00005890
     2/,3(15X,20(I3,' ,'),/),/)                                         00005900
C                                                                       00005910
      IF (NNRT.NE.0) READ (5,304) (NNR(I),I=1,NNRT)                     00005920
      IF (NNRT.NE.0) WRITE (6,309) NNRT,(NNR(I),I=1,NNRT)               00005930
  309 FORMAT (/,15X,'TOTAL NUMBER OF NODES FIXED IN THE ',              00005940
     1'RADIAL DIRECTION ARE',I5,' AND THEY ARE AS :',/,                 00005950
     23(15X,20(I3,' ,'),/),/)                                           00005960
  304 FORMAT  (10I5)                                                    00005970
C                                                                       00005980
      IF (IPLANE.NE.0)  READ (5,102) NNRE,(NRE(I),I=1,NNRE)             00005990
      IF (IPLANE.NE.0) WRITE (6,214) NNRE,(NRE(I),I=1,NNRE)             00006000
  214 FORMAT (/,15X,'TOTAL NUMBER OF NODES FIXED IN LONGIT',            00006010
     1'UDINAL DIRECTION OF RIGHT HAND END IS ',I5,' AND TH',            00006020
     3'EY ARE AS:',/,2(15X,20(I3,' ,'),/),/)                            00006030
C                                                                       00006040
      READ  (5,100)  TINIT , Q2 , Q3 , Q4 , AXIALF                      00006050
C                                                                       00006060
      IF  (AXIALF.EQ.0.D0)  WRITE  (6,220)                              00006070
  220 FORMAT  (/,15X,'THERE IS NO AXIAL FORCE ON THE BODY',/)           00006080
      IF  (AXIALF.NE.0.D0)  WRITE  (6,221)  AXIALF                      00006090
  221 FORMAT  (/,15X,'AXIAL FORCE = ',F10.4,/)                          00006100
C                                                                       00006110
C                                                                       00006120
      IF  (IEXT.NE.0)  GO  TO  52                                       00006130
      IF  (Q4.LT.0.D0)  GO  TO  57                                      00006140
C                                                                       00006150
CC                                                                      00006160
CC       IF  CONVECTION   BDRY.  OUTSIDE   Q2 > 0.D0                    00006170
CC       IF  INSULATED    BDRY.  OUTSIDE   Q2 = 0.D0                    00006180
```

```
C                                                                       00006190
C                                                                       00006200
C       IF  CONVECTION   BDRY.   INSIDE   Q3 > 0.D0                     00006210
C       IF  INSULATED    BDRY.   INSIDE   Q3 = 0.D0                     00006220
C                                                                       00006230
C                                                                       00006240
C       Q4.GT.0.D0  TEMPERATURE PROBLEM IS TO BE SOLVED ONLY            00006250
C       Q4 = 0.D0   STRESS PROBLEM IS TO BE SOLVED ALSO.                00006260
C       Q4.LT.0.D0  STRESS PROBLEM IS TO BE SOLVED ONLY.                00006270
C                                                                       00006280
C                                                                       00006290
      IF  (Q2. GT . 0.D0)  GO  TO  1                                    00006300
      GO  TO  3                                                         00006310
    1 CONTINUE                                                          00006320
C                                                                       00006330
      READ  (5,103)  HTCO , TEMPO , NCFOT , NRAMPO                      00006340
      N = NCFOT                                                         00006350
      READ  (5,102)  (NCFO(I), I = 1 , NCFOT)                           00006360
      NRPPO = NRAMPO + 1                                                00006370
C                                                                       00006380
C                                                                       00006390
      READ  (5,104)  ((BDRYO(I,J),J=1,3),BDRYO(I,5),I = 2,NRPPO)        00006400
C                                                                       00006410
C     BDRYO(I,1) IS THE INITIAL TIME, TO BE READ IN FOR                 00006420
C     EACH RAMP SEGMENT.                                                00006430
C     BDRYO(I,2) IS THE INITIAL TEMPERATURE , TO BE READ IN.            00006440
C     BDRYO(I,3) IS THE FINAL TEMPERATURE TO BE READ IN IF IT DIFFERS   00006450
C     FROM THE NEXT INITIAL TEMPERATURE,                                00006460
C     BDRYO(I,4) IS THE SLOPE , WHICH IS CALCULATED BY THE              00006470
C     PROGRAM.                                                          00006480
C     BDRYO(I,5) IS THE VELOCITY OF THE FLUID TO BE READ IN             00006490
C     IF  IT DIFFERS FROM THE PREVIOUS RAMP VELOCITY                    00006500
C                                                                       00006510
C                                                                       00006520
      WRITE  (6,2021) HTCO                                              00006530
C                                                                       00006540
C                                                                       00006550
      DO  4  I = 1 , NCFOT                                              00006560
      FT2(I) = TEMPO                                                    00006570
    4 FT1(I) = TEMPO                                                    00006580
C                                                                       00006590
C                                                                       00006600
      WRITE  (6,201)  (NCFO(I),FT1(I),I=1,NCFOT)                        00006610
C                                                                       00006620
C                                                                       00006630
      BDRYO(1,1) = 0.D0                                                 00006640
      BDRYO(1,2) = TEMPO                                                00006650
      BDRYO(1,4) = 0.D0                                                 00006660
```

```
      BDRYO(1,5) = 0.D0                                                    00006670
      BDRYO(NRPPO,4) = 0.D0                                                00006680
      BDRYO(NRPPO+1,1) = 999999.D0                                         00006690
C                                                                          00006700
      DO  5  I = 2 , NRAMPO                                                00006710
      IF  (BDRYO(I,3).EQ.0.D0)  BDRYO(I,3) = BDRYO(I+1,2)                  00006720
      IF  (BDRYO(I,5).EQ.0.D0)  BDRYO(I,5) = BDRYO(I-1,5)                  00006730
      DUM = BDRYO(I,3) - BDRYO(I,2)                                        00006740
    5 BDRYO(I,4) = DUM/(BDRYO(I+1,1) - BDRYO(I,1))                         00006750
C                                                                          00006760
C                                                                          00006770
      WRITE  (6,202)  (BDRYO(I,1),BDRYO(I,2),BDRYO(I+1,1),                00006780
     1BDRYO(I,3),BDRYO(I,4),BDRYO(I,5),I = 2,NRPPO)                        00006790
C                                                                          00006800
      IF  (Q1.EQ.BRIT)  GO  TO  20                                         00006810
      DO  21  I = 2 , NRPPO                                                00006820
   21 BDRYO(I,5) = BDRYO(I,5) * 100.D0                                     00006830
      GO  TO  18                                                           00006840
   20 DO  22  I = 2 , NRPPO                                                00006850
   22 BDRYO(I,5) = BDRYO(I,5) * 12.D0                                      00006860
C                                                                          00006870
      GO  TO  18                                                           00006880
C                                                                          00006890
C                                                                          00006900
    3 WRITE  (6,1000)                                                      00006910
   18 CONTINUE                                                             00006920
C                                                                          00006930
      IF  (Q3. GT .0D0)  GO  TO  7                                         00006940
      WRITE  (6,1001)                                                      00006950
      GO  TO  9                                                            00006960
C                                                                          00006970
    7 READ  (5,103)  HTCI , TEMPI , NCFIT , NRAMPI                         00006980
      READ  (5,102)  (NCFI(I) , I = 1 , NCFIT)                             00006990
C                                                                          00007000
      NRPPI = NRAMPI + 1                                                   00007010
      READ  (5,104)  ((BDRYI(I,J),J=1,3),BDRYI(I,5),I=2,NRPPI)             00007020
C                                                                          00007030
C                                                                          00007040
      WRITE  (6,2022)  HTCI                                                00007050
C                                                                          00007060
C                                                                          00007070
      DO  10  I = 1 , NCFIT                                                00007080
      K = N + I                                                            00007090
      FT1(K) = TEMPI                                                       00007100
   10 FT2(K) = TEMPI                                                       00007110
C                                                                          00007120
C                                                                          00007130
      DO  19  I = 1 , NCFIT                                                00007140
```

```
      K = N + I                                                          00007150
   19 WRITE  (6,201)  NCFI(I) , FT1(K)                                   00007160
C                                                                        00007170
C                                                                        00007180
      BDRYI(1,1) = 0.D0                                                  00007190
      BDRYI(1,2) = TEMPI                                                 00007200
      BDRYI(1,4) = 0.D0                                                  00007210
      BDRYI(1,5) = 0.D0                                                  00007220
      BDRYI(NRPPI,4) = 0.D0                                              00007230
      BDRYI(NRPPI+1,1) = 999999.D0                                       00007240
C                                                                        00007250
C                                                                        00007260
C                                                                        00007270
      DO  11  I = 2 , NRAMPI                                             00007280
      IF  (BDRYI(I,3).EQ.0.D0)  BDRYI(I,3) = BDRYI(I+1,2)                00007290
      IF  (BDRYI(I,5).EQ.0.D0)  BDRYI(I,5) = BDRYI(I-1,5)                00007300
      DUM = BDRYI(I,3) - BDRYI(I,2)                                      00007310
   11 BDRYI(I,4) = DUM / (BDRYI(I+1,1)-BDRYI(I,1))                       00007320
C                                                                        00007330
      WRITE  (6,202)  (BDRYI(I,1),BDRYI(I,2),BDRYI(I+1,1),              00007340
     1BDRYI(I,3),BDRYI(I,4),BDRYI(I,5),I=2,NRPPI)                        00007350
C                                                                        00007360
      IF  (Q1.EQ.BRIT)  GO  TO  23                                       00007370
      DO  24  I = 2 , NRPPI                                              00007380
   24 BDRYI(I,5) = BDRYI(I,5) * 100.D0                                   00007390
      GO  TO  9                                                          00007400
   23 DO  25  I = 2 , NRPPI                                              00007410
   25 BDRYI(I,5) = BDRYI(I,5) * 12.D0                                    00007420
C                                                                        00007430
    9 READ  (5,106)  DTI , TIME1 , EVERY , IVEC , INTP                   00007440
C                                                                        00007450
C                                                                        00007460
      TS = TIME1 + EVERY * (IVEC-1)                                      00007470
C                                                                        00007480
      WRITE  (6,215)  DTI                                                00007490
  215 FORMAT  (15X,'STEP SIZE OF TIME INTEGRATION IS ',F10.4,/)          00007500
C                                                                        00007510
      DO  14  I = 1,NNT                                                  00007520
   14 TEMP(I) = TINIT                                                    00007530
      WRITE  (6,204)  TINIT                                              00007540
      WRITE  (6,205)  TREF                                               00007550
      WRITE  (6,206)  IVEC                                               00007560
C                                                                        00007570
      WRITE  (6,208)  TIME1                                              00007580
      WRITE  (6,209)  EVERY                                              00007590
      IF  (INTP.EQ.0)  GO  TO  40                                        00007600
      WRITE  (6,210)  INTP                                               00007610
      GO  TO  41                                                         00007620
```

```
   40 WRITE  (6,211)                                                    00007630
   41 CONTINUE                                                          00007640
C                                                                       00007650
  208 FORMAT  (15X,'THE FIRST TEMP. VECTOR IS KEPT AT TIME ',           00007660
     1F10.4,/)                                                          00007670
C                                                                       00007680
  209 FORMAT  (15X,'THEN TEMP. VECTORS ARE KEPT AT EVERY.',             00007690
     13X,F10.4,'   SECONDS',/)                                          00007700
C                                                                       00007710
  210 FORMAT  (15X,'TEMPERATURES ARE PRINTED FOR EVERY',5X,I5,          00007720
     1'  INTERVAL OF TIME',//)                                          00007730
C                                                                       00007740
  211 FORMAT  (15X,'TEMPERATURES ARE NOT PRINTED',/)                    00007750
C                                                                       00007760
CC                                                                      00007770
C                                                                       00007780
      KK = 1                                                            00007790
      NPASS = TS / DTI                                                  00007800
      NPASS = NPASS + NPASS/10                                          00007810
C                                                                       00007820
C                                                                       00007830
  201 FORMAT (15X,'NODE',I5,2X,'HAS CONTACT WITH FLUID AT',2X,F10.4,    00007840
     13X,'DEGREE TEMPERATURE')                                          00007850
C                                                                       00007860
  202 FORMAT (//,15X,'ENTRY TEMPERATURE AND TIME RELATION',             00007870
     1/,15X,34('*'),//,15X,'INITIAL TIME',5X,'INITIAL ',               00007880
     2'TEMPERATURE',10X,'FINAL TIME',5X,'FINAL TEMPERATURE',           00007890
     310X,'SLOPE',5X,'VELOCITY',//,(17X,F10.4,10X,F10.4,               00007900
     414X,F10.2,8X,F10.4,10X,F10.4,5X,F10.4))                          00007910
C                                                                       00007920
CC                                                                      00007930
  203 FORMAT  (/,15X,'THE SYSTEM IS ROTATING ',F10.5,' REV',            00007940
     1'OLUTIONS PER MINUTE',/)                                          00007950
C                                                                       00007960
CC                                                                      00007970
  204 FORMAT (//,15X,'INITIAL TEMPERATURE OF THE SOLID IS',             00007980
     1'  ',F10.3,/)                                                    00007990
  205 FORMAT  (15X,'REFERENCE  TEMPERATURE IS',12X,F10.3,/)             00008000
  206 FORMAT  (15X,'NUMBER OF STORED TEMPERATURE VECTORS IS',I6,/)      00008010
C                                                                       00008020
  207 FORMAT  (15X,'INSIDE PRESSURE IS',19X,F10.5,/)                    00008030
C                                                                       00008040
CC                                                                      00008050
 1000 FORMAT (//,15X,'OUTSIDE BDRY. COND. IS INSULATED',/)              00008060
 1001 FORMAT (//,15X,'INSIDE  BDRY. COND. IS INSULATED',/)              00008070
C                                                                       00008080
C                                                                       00008090
 2021 FORMAT (//,15X,'OUTSIDE BDRY. COND. IS CONVECTION',//,            00008100
```

```
     115X,'AND THE HEAT TRANSFER COEF. IS ',1PD14.6,//)                  00008110
 2022 FORMAT (//,15X,'INSIDE  BDRY. COND. IS CONVECTION',//,             00008120
     115X,'AND THE HEAT TRANSFER COEF. IS ',1PD14.6,//)                  00008130
C                                                                        00008140
C                                                                        00008150
   52 IF  (IEXT.EQ.0)  GO  TO  54                                        00008160
      NNT1 = NNT + 1                                                     00008170
      READ  (5,109)  ((STOR(I,J),J=1,NNT1),I=1,IEXT)                     00008180
      IVEC = IEXT                                                        00008190
      WRITE  (6,212)  IEXT                                               00008200
  212 FORMAT  (/,15X,I5,' TEMPERATURE VECTORS ARE READ IN AND',          00008210
     1/,15X,'    TEMPERATURES ARE IN ORDER  OF GLOBAL NODE ',            00008220
     2'NUMBERS :',/)                                                     00008230
C                                                                        00008240
      DO  56  I = 1 , IEXT                                               00008250
      DO  56  J = 1 , NNT                                                00008260
   56 WRITE  (6,213)  J , STOR(I,J)                                      00008270
  213 FORMAT    (15X,'NODE(',I3,')  ',F10.3)                             00008280
   54 CONTINUE                                                           00008290
      IF  (Q1.EQ.BRIT)  GO  TO  60                                       00008300
C                                                                        00008310
      GO  TO  62                                                         00008320
C                                                                        00008330
   60 IF  (Q2.GT.0.D0)  HTCO = HTCO / 518400.D0                          00008340
      IF  (Q3.GT.0.D0)  HTCI = HTCI / 518400.D0                          00008350
C                                                                        00008360
      GO  TO  58                                                         00008370
C                                                                        00008380
   57 IVEC = 1                                                           00008390
      STOR(1,NNT+1) = 0.D0                                               00008400
      DO  59  J = 1 , NNT                                                00008410
   59 STOR(1,J) = TREF                                                   00008420
   58 CONTINUE                                                           00008430
   62 OMEGA = OMEGA * (PAI / 30.D0)                                      00008440
C                                                                        00008450
      BIG = 10.D0**20                                                    00008460
  100 FORMAT  (5F10.4)                                                   00008470
  101 FORMAT (2F10.4,5I5)                                                00008480
C                                                                        00008490
  102 FORMAT  (12I5)                                                     00008500
  103 FORMAT  (D16.4,F10.4,2I5)                                          00008510
C                                                                        00008520
  104 FORMAT  (4F10.4)                                                   00008530
  106 FORMAT  (3F10.4,2I5)                                               00008540
C                                                                        00008550
  108 FORMAT  (6F10.4)                                                   00008560
  109 FORMAT  (6F10.3)                                                   00008570
C                                                                        00008580
   RETURN                                                                00008590
   END                                                                   00008600
```

```
C                                                                       00008610

      SUBROUTINE  PLOTZR(X , Y , N)                                     00008620
C                                                                       00008630
CC                                                                      00008640
CC       ***************************************************************00008650
CC                                                                      *00008660
CC       SUBROUTINE PLOTZR WILL PRODUCE A PLOT OF THE NODAL             *00008670
CC       POINTS OF ANY TWO DIMENSIONAL GEOMETRY . THE NODES             *00008680
CC       ARE REPRESENTED BY '*' , AND THE APPROPRIATE NODAL             *00008690
CC       POINT NUMBERS ARE PRINTED BELOW EACH NODE.                     *00008700
CC       PLOTZR(X,Y,N) ; X IS HORIZONTAL COORDINATE                     *00008710
CC                       Y IS VERTICAL   COORDINATE                     *00008720
CC                       N IS NUMBER OF NODES. (LESS THAN 1000)         *00008730
CC                                                                      *00008740
CC       ***************************************************************00008750
CC                                                                      00008760
C                                                                       00008770
      DIMENSION  X(N),Y(N),IDUM(104),JDUM(104),NU(10)                   00008780
      DATA  NU/'1','2','3','4','5','6','7','8','9','0'/                00008790
      DATA  ISTAR/'*'/ , IBLANK/' '/ , IDOT/'.'/                        00008800
C                                                                       00008810
      DO  20  I = 1 , 104                                               00008820
   20 IDUM(I) = IDOT                                                    00008830
CC                                                                      00008840
CC                                                                      00008850
      WRITE  (6,85)  IDUM                                               00008860
   85 FORMAT  (1H1,  /,45X,'GRAPHICAL PRESENTATION OF NODAL ',          00008870
     1'POINTS.',/,48X,'Z  HORIZONTAL       R VERTICAL',/,20X,          00008880
     2104A1)                                                            00008890
C                                                                       00008900
CC                                                                      00008910
      XMAX = -1.0E 15                                                   00008920
      YMAX = -1.0E 15                                                   00008930
      XMIN = - XMAX                                                     00008940
      YMIN = - YMAX                                                     00008950
C                                                                       00008960
CC                                                                      00008970
      DO  1  I = 1 , N                                                  00008980
      IF  (X(I).GT.XMAX)    XMAX = X(I)                                 00008990
      IF  (X(I).LT.XMIN)    XMIN = X(I)                                 00009000
      IF  (Y(I).GT.YMAX)    YMAX = Y(I)                                 00009010
      IF  (Y(I).LT.YMIN)    YMIN = Y(I)                                 00009020
    1 CONTINUE                                                          00009030
C                                                                       00009040
CC                                                                      00009050
      DIFX = XMAX - XMIN                                                00009060
```

```
      DIFY = YMAX - YMIN                                                00009070
      FACTX = 100.0 / DIFX                                              00009080
      FACTY = 70.0 / DIFY                                               00009090
C                                                                       00009100
      DO  2  I = 1 , N                                                  00009110
      X(I) = FACTX * (X(I) - XMIN)                                      00009120
      Y(I) = 1.0 + FACTY * (Y(I) - YMIN)                                00009130
    2 CONTINUE                                                          00009140
C                                                                       00009150
      WRITE  (6,87)  YMAX                                               00009160
C                                                                       00009170
C                                                                       00009180
      JDUM(1) = IDOT                                                    00009190
      IDUM(103) = IBLANK                                                00009200
      K = 71                                                            00009210
      DO  9  I = 1 , 71                                                 00009220
      JDUM(104) = IDOT                                                  00009230
      DO  6  J = 2 , 103                                                00009240
      JDUM(J) = IBLANK                                                  00009250
    6 IDUM(J) = IBLANK                                                  00009260
C                                                                       00009270
C                                                                       00009280
      DO  7  J = 1 , N                                                  00009290
      M = Y(J)                                                          00009300
      IF  (M.EQ.K)  GO  TO  8                                           00009310
      GO  TO  7                                                         00009320
    8 NUM = X(J) + 2                                                    00009330
      IDUM(NUM) = ISTAR                                                 00009340
C                                                                       00009350
      K1 = J/100                                                        00009360
      IF  (K1.NE.0)  GO  TO  15                                         00009370
      K1 = J/10                                                         00009380
      IF  (K1.NE.0)  GO  TO  10                                         00009390
      JDUM(NUM) = NU(J)                                                 00009400
      GO  TO  7                                                         00009410
C                                                                       00009420
   10 K2 = J - 10*K1                                                    00009430
      JDUM(NUM) = NU(K1)                                                00009440
      IF  (K2.EQ.0)  K2 = 10                                            00009450
      JDUM(NUM+1) = NU(K2)                                              00009460
      GO  TO  7                                                         00009470
C                                                                       00009480
   15 K3 = J - 100*K1                                                   00009490
      K2 = K3 / 10                                                      00009500
      K3 = K3 - 10*K2                                                   00009510
      JDUM(NUM) = NU(K1)                                                00009520
      IF  (K2.EQ.0)  K2 = 10                                            00009530
      JDUM(NUM+1) = NU(K2)                                              00009540
```

```
      IF  (K3.EQ.0)  K3 = 10                                          00009550
      JDUM(NUM+2) = NU(K3)                                            00009560
C                                                                     00009570
    7 CONTINUE                                                        00009580
C                                                                     00009590
C                                                                     00009600
      WRITE  (6,88)  IDUM                                             00009610
      WRITE  (6,88)  JDUM                                             00009620
      K = K - 1                                                       00009630
    9 CONTINUE                                                        00009640
C                                                                     00009650
C                                                                     00009660
      WRITE  (6,89)  YMIN                                             00009670
C                                                                     00009680
C                                                                     00009690
      DIF = (XMA -XMIN) / 4.0                                         00009700
      X2 = XMIN + DIF                                                 00009710
      X3 = X2 + DIF                                                   00009720
      X4 = X3 + DIF                                                   00009730
      WRITE  (6,90)  XMIN , X2 , X3 , X4 , XMAX                       00009740
C                                                                     00009750
   87 FORMAT  (3X,'MAX.R=',F10.5)                                     00009760
   89 FORMAT  (3X,'MIN.R=',F10.5)                                     00009770
   88 FORMAT  (20X,104A1)                                             00009780
   90 FORMAT  (20X,'.+',24('.'),'+',24('.'),'+',24('.'),'+',          00009790
     124('.'),'+..',/,16X,4(F10.5,15X),F10.5,'   Z')                  00009800
C                                                                     00009810
C                                                                     00009820
      WRITE  (6,555)                                                  00009830
  555 FORMAT  (///,40X,'NOTE :',/,45X,'IF THE NODES ARE ',            00009840
     1'TOO CLOSED TO EACH OTHER',/,45X,'A NODE MAY NOT BE ',          00009850
     2'PRINTED , A NODE NUMBER',/,45X,'MAY BE MISPRINTED ',',         00009860
     3'OR BOTH.')                                                     00009870
      RETURN                                                          00009880
      END                                                             00009890
```

```
C                                                                       00009900

      SUBROUTINE  PLOT(X,Y,LL)                                          00009910
C                                                                       00009920
      DIMENSION  X(75),Y(75),IDUM(102)                                  00009930
      DATA ISTAR/'*'/ , IPLUS/'+'/ , IBLANK/' '/ , IDOT/'.'/            00009940
C                                                                       00009950
CCC                                                                     00009960
CCC      ************************************************************** 00009970
CCC                                                                   * 00009980
CCC      SUBROUTINE PLOT(X,Y,LL) WILL PLOT THE VARIATION OF           ** 00009990
CCC      THE ENTRY TEMPERATURE VERSUS TIME AS 75 POINTS FOR           ** 00010000
CCC      THE FIRST 150 TIME INCREMENTS,IF(TEMPERATURE.GE.0.)          ** 00010010
CCC           X  IS TIME.                                             ** 00010020
CCC           Y  IS TEMPERATURE.                                      ** 00010030
CCC           LL=0  OUTSIDE FLOW.                                     ** 00010040
CCC           LL=1  INSIDE  FLOW.                                     ** 00010050
CCC                                                                   * 00010060
CCC      ************************************************************** 00010070
CC                                                                      00010080
C                                                                       00010090
      DO  1  I = 1 , 75                                                 00010100
      IF  (Y(I).LT.0.0)  GO  TO  50                                     00010110
    1 CONTINUE                                                          00010120
      DO  5  I = 1 , 102                                                00010130
    5 IDUM(I) = IDOT                                                    00010140
C                                                                       00010150
CC                                                                      00010160
      YMAX = -1.0E 18                                                   00010170
      YMIN = - YMAX                                                     00010180
      DO  2  I = 1 , 75                                                 00010190
      IF  (Y(I). GT .YMAX)  YMAX = Y(I)                                 00010200
      IF  (Y(I). LT .YMIN)  YMIN = Y(I)                                 00010210
    2 CONTINUE                                                          00010220
C                                                                       00010230
C                                                                       00010240
      RAT = 1.0 + (100.0 * YMIN) / YMAX                                 00010250
      NUM = RAT                                                         00010260
      IDUM(NUM) = IPLUS                                                 00010270
C                                                                       00010280
C                                                                       00010290
      IF  (LL.EQ.0)  WRITE  (6,66)  IDUM                                00010300
   66 FORMAT (1H1, / ,42X,'OUTSIDE FLUID ENTRY TIME TEMPERA'            00010310
     1'TURE PLOT',/,42X,41('*'), / ,7X,'TIME',5X,'TEMPERATURE',         00010320
     290X,'TEMPERATURE',/,29X,102A1)                                    00010330
C                                                                       00010340
C                                                                       00010350
```

```
      IF  (LL.NE.0)  WRITE  (6,77)  IDUM                                00010360
   77 FORMAT (1H1,' ',42X,' INSIDE FLUID ENTRY TIME TEMPERA'            00010370
     1'TURE PLOT',/,43X,40('*'), / ,7X,'TIME',5X,'TEMPERATURE',         00010380
     290X,'TEMPERATURE',/,29X,102A1)                                    00010390
C                                                                       00010400
C                                                                       00010410
      DO  3  I = 1 , 101                                                00010420
    3 IDUM(I) = IBLANK                                                  00010430
C                                                                       00010440
C                                                                       00010450
      DO  4  I = 1 , 75                                                 00010460
      NUM = 1 + 100 * Y(I)/YMAX                                         00010470
      IDUM(NUM) = ISTAR                                                 00010480
      WRITE  (6,88)  X(I),Y(I),IDUM                                     00010490
      IDUM(NUM) = IBLANK                                                00010500
    4 CONTINUE                                                          00010510
   88 FORMAT  (5X,F8.2,4X,F10.3,'  .',102A1)                            00010520
C                                                                       00010530
C                                                                       00010540
      DO  6  I = 1 , 101                                                00010550
    6 IDUM(I) = IDOT                                                    00010560
      WRITE  (6,99) IDUM                                                00010570
   99 FORMAT  (29X,102A1)                                               00010580
   50 RETURN                                                            00010590
      END                                                               00010600
C                                                                       00010610
```

```
      SUBROUTINE  LLT  (N,A,WL)                                          00010620
C                                                                        00010630
CCC                                                                      00010640
CCC                                                                      00010650
CCC      ************************************************************* ***00010660
CCC                                                                      *00010670
CCC      THIS SUBROUTINE WILL DECOMPOSE ANY  N BY N  REAL                *00010680
CCC      SYMMETRIC MATRIX A INTO A LOWER TRIANGULAR MATRIX               *00010690
CCC      'WL' OF THE SAME ORDER N , SUCH THAT THE PRODUCT                *00010700
CCC      OF 'WL' AND ITS TRANSPOSE IS THE GIVEN MATRIX  A.               *00010710
CCC      (THIS IS CHOLESKY DECOMPOSITION.)                               *00010720
CCC                                                                      *00010730
CCC      ************************************************************** 00010740
C                                                                        00010750
                                                                         00010760
      IMPLICIT  REAL *8 (A-H,O-Z)                                        00010770
      DIMENSION  A(N,N),WL(N,N)                                          00010780
      DO   7   I=1,N                                                     00010790
      DO   8   J=1,N                                                     00010800
    8 WL(I,J)=0.0D0                                                      00010810
    7 CONTINUE                                                           00010820
      WL(1,1)=DSQRT(A(1,1))                                              00010830
      DO   1   I=2,N                                                     00010840
    1 WL(I,1)=A(I,1)/WL(1,1)                                             00010850
      N1=N-1                                                             00010860
      M=3                                                                00010870
      DO   2   I=2,N                                                     00010880
      I1=I-1                                                             00010890
      DO   3   J=1,I1                                                    00010900
    3 WL(I,I)=WL(I,I)+WL(I,J)**2                                         00010910
      WL(I,I)=A(I,I)-WL(I,I)                                             00010920
      WL(I,I)=DSQRT(WL(I,I))                                             00010930
      IF  (M.GT.N) GO TO   9                                             00010940
      DO   5   II=M,N                                                    00010950
      DO   6   L=1,I1                                                    00010960
    6 WL(II,I)=WL(II,I)+WL(I,L)*WL(II,L)                                 00010970
    5 WL(II,I)=(A(II,I)-WL(II,I))/WL(I,I)                                00010980
    2 M=M+1                                                              00010990
    9 RETURN                                                             00011000
      END                                                                00011010
```

```
C                                                                       00011020

      SUBROUTINE  CANDY                                                 00011030
C                                                                       00011040
CCCCC                                                                   00011050
CCCCC  ***************************************************************  00011060
CCCCC                                                                 *  00011070
CCCCC  SUBROUTINE CANDY WILL CALCULATE CAPACITANCE MATRIX             *  00011080
CCCCC  C AND ADMITTANCE MATRIX Y IN THE BANDED FORM  THEN             *  00011090
CCCCC  THE ADDITIONAL YSTAR MATRIX IS EVALUATED AND ADDED             *  00011100
CCCCC  TO THE Y MATRIX .  A AND G  MATRICES ARE EVALUATED             *  00011110
CCCCC  AND PLACED IN THE  C AND Y  RESPECTIVELY.                      *  00011120
CCCCC  FINALLY,A CHOLESKY DECOMPOSITION IS PERFORMED ON C.            *  00011130
CCCCC                                                                 *  00011140
CCCCC  ***************************************************************  00011150
C                                                                       00011160
C                                                                       00011170
      IMPLICIT  REAL * 8 (A-H,O-Z)                                      00011180
C                                                                       00011190
      COMMON /ONE/ A(2,2),B(4,16),COPY(37,298),D(4,4),DL(4,4),          00011200
     1DNZR(2,8),ESM(16,16),F(24,298),SSM(298,66),STOR(20,150)           00011210
C                                                                       00011220
      COMMON /TWO/ AL(5),DENS(5),DETA(8),DXI(8),E(5),EPSO(4),           00011230
     1FE(16),GC(4),GX(4),POI(5),R(149),SHP(8),SHT(5),TK(5),Z(149)       00011240
C                                                                       00011250
      COMMON /TRE/ AXIALF,DTI,EVERY,FORS1,OMEGA,HTCI,HTCO,              00011260
     1POSI,PRES,Q1,Q2,Q3,SHMAX,SMAX,TIME,TIME1,TINIT,TMAXM,TREF,        00011270
     2TMAXO,RMAXM,RMAXO,ZMAXM,ZMAXO,Q4,BIG                              00011280
C                                                                       00011290
      COMMON /FOR/ NN(40,9)                                             00011300
C                                                                       00011310
      COMMON /FIV/ NCFI(37),NCFO(37),NNR(37),NRE(37),NNL(37),           00011320
     1NPN(37)                                                           00011330
      COMMON /SIX/ IBAND,IEXT,INTP,IVEC,JVEC,KK,LOAD,NCFIT,NCFOT,       00011340
     1NCHECK,NET,NDF,NGP,NME,NNLT,NNRT,NPASS,NPROB,NRPPI,NRPPO,         00011350
     3NNT,NNRE,NPNT,IPLANE,NBAND                                        00011360
C                                                                       00011370
      EQUIVALENCE  (C,SSM) , (Y,SSM(1,18)) , (T,SSM(1,35)) ,            00011380
     1(BTE,SSM(1,36)) , (TEMP,SSM(1,37)) , (V,SSM(1,38)) ,             00011390
     2(FT1,SSM(1,39)) , (FT2,SSM(1,40)) , (CE,SSM(1,41)) ,             00011400
     3(YE,SSM(1,42)) , (OFF1,SSM(1,43)) , (OFF2,SSM(1,44)) ,           00011410
     4(DIAG,SSM(1,45)) , (BDRYI,SSM(1,46)) , (BDRYO,SSM(1,47)) ,       00011420
     5(ENTII,SSM(1,48)) , (ENTIO,SSM(1,49)), (ENTEI,SSM(1,50)),        00011430
     6(ENTEO,SSM(1,51)),(VOL,SSM(1,52)),(VOLO,SSM(1,53))               00011440
C                                                                       00011450
      DIMENSION  C(149,33),Y(149,33),T(149),BTE(149),TEMP(149),         00011460
     1V(149),FT1(90),FT2(90),CE(8,8),YE(8,8),OFF1(37),OFF2(37),         00011470
```

```
     2DIAG(37),BDRYI(16,5),BDRYO(16,5),VOL(36),VOLO(36),ENTEO(75),      00011480
     3ENTEI(75),ENTIO(75),ENTII(75)                                     00011490
C                                                                       00011500
      DO  12  I = 1 , NNT                                               00011510
      V(I) = 0.DO                                                       00011520
      DO  12  J = 1 , NBAND                                             00011530
      C(I,J) = 0.DO                                                     00011540
      Y(I,J) = 0.DO                                                     00011550
   12 CONTINUE                                                          00011560
C                                                                       00011570
C                                                                       00011580
      PAI = 3.141592653589793                                           00011590
      DO  25  L = 1 , NET                                               00011600
C                                                                       00011610
C                                                                       00011620
      M1 = NN(L,9)                                                      00011630
      FACTC = 2.DO * PAI * DENS(M1) * SHT(M1)                           00011640
      FACTY = 2.DO * PAI * TK(M1)                                       00011650
C                                                                       00011660
CC                                                                      00011670
C                                                                       00011680
      DO  7  I = 1 , NNE                                                00011690
      DO  7  J = 1 , NNE                                                00011700
      CE(I,J) = 0.DO                                                    00011710
    7 YE(I,J) = 0.DO                                                    00011720
C                                                                       00011730
CC                                                                      00011740
C                                                                       00011750
      DO  9  N = 1 , NGP                                                00011760
C                                                                       00011770
      ETA = GX(N)                                                       00011780
C                                                                       00011790
      DO  9  M = 1 , NGP                                                00011800
C                                                                       00011810
      XI = GX(M)                                                        00011820
C                                                                       00011830
CC                                                                      00011840
CC       SHAPE FUNCTIONS.                                               00011850
CC                                                                      00011860
C                                                                       00011870
      SHP(1) = .25D0 * (1.D0-XI) * (1.D0-ETA) * (-XI-ETA-1.D0)          00011880
      SHP(2) = .5D0 * (1.D0-XI**2) * (1.D0-ETA)                         00011890
      SHP(3) = .25D0 * (1.D0+XI) * (1.D0-ETA) * (XI-ETA-1.D0)           00011900
      SHP(4) = .5D0 * (1.D0-ETA**2) * (1.D0+XI)                         00011910
      SHP(5) = .25D0 * (1.D0+XI) * (1.D0+ETA) * (XI+ETA-1.D0)           00011920
      SHP(6) = .5D0 * (1.D0-YI**2) * (1.D0+ETA)                         00011930
      SHP(7) = .25D0 * (1.D0-XI) * (1.D0+ETA) * (-XI+ETA-1.D0)          00011940
      SHP(8) = .5D0 * (1.D0-ETA**2) * (1.D0-XI)                         00011950
```

```
C                                                                       00011960
C                                                                       00011970
C                                                                       00011980
C     DERIVATIVES OF THE SHAPE FUNCTIONS WITH RESPECT TO XI.            00011990
C                                                                       00012000
C                                                                       00012010
      DXI(1)     = .25D0 * (1.D0-ETA) * (2.D0*XI+ETA)                   00012020
      DXI(2)     = -XI * (1.D0-ETA)                                     00012030
      DXI(3)     = .25D0 * (1.D0-ETA) * (2.D0*XI-ETA)                   00012040
      DXI(4)     = .5D0 * (1.D0-ETA**2)                                 00012050
      DXI(5)     = .25D0 * (1.D0+ETA) * (2.D0*XI+ETA)                   00012060
      DXI(6)     = -XI * (1.D0+ETA)                                     00012070
      DXI(7)     = .25D0 * (1.D0+ETA) * (2.D0*XI-ETA)                   00012080
      DXI(8)     = -.5D0 * (1.D0-ETA**2)                                00012090
C                                                                       00012100
C                                                                       00012110
C     DERIVATIVES OF THE SHAPE FUNCTIONS WITH RESPECT TO ETA.           00012120
C                                                                       00012130
C                                                                       00012140
      DETA(1)    = .25D0 * (1.D0-XI) * (2.D0*ETA+XI)                    00012150
      DETA(2)    = -.5D0 * (1.D0-XI**2)                                 00012160
      DETA(3)    = .25D0 * (1.D0+XI) * (2.D0*ETA-XI)                    00012170
      DETA(4)    = -ETA * (1.D0+XI)                                     00012180
      DETA(5)    = .25D0 * (1.D0+XI) * (2.D0*ETA+XI)                    00012190
      DETA(6)    = .5D0 * (1.D0-XI**2)                                  00012200
      DETA(7)    = .25D0 * (1.D0-XI) * (2.D0*ETA-XI)                    00012210
      DETA(8)    = -ETA * (1.D0-XI)                                     00012220
C                                                                       00012230
C                                                                       00012240
C        CONSTRUCTION OF THE JACOBIAN MATRIX.                           00012250
C                                                                       00012260
C                                                                       00012270
      DO  5  I = 1 , 2                                                  00012280
      DO  5  J = 1 , 2                                                  00012290
    5 AJ(I,J) = 0.D0                                                    00012300
C                                                                       00012310
      DO  22  I = 1 , NNE                                               00012320
      AJ(1,1) = AJ(1,1) + DXI(I) * Z(NN(L,I))                           00012330
      AJ(1,2) = AJ(1,2) + DXI(I) * R(NN(L,I))                           00012340
      AJ(2,1) = AJ(2,1) + DETA(I) * Z(NN(L,I))                          00012350
      AJ(2,2) = AJ(2,2) + DETA(I) * R(NN(L,I))                          00012360
   22 CONTINUE                                                          00012370
C                                                                       00012380
C                                                                       00012390
      DET = AJ(1,1) * AJ(2,2) - AJ(1,2) * AJ(2,1)                       00012400
      DEI = 1.D0 / DET                                                  00012410
C                                                                       00012420
C        CONSTRUCTION OF INVERSE OF JACOBIAN                            00012430
```

```
C                                                                       00012440
C                                                                       00012450
      DUM = AJ(1,1) * DEI                                               00012460
      AJ(1,1) =  AJ(2,2) * DEI                                          00012470
      AJ(1,2) = -AJ(1,2) * DEI                                          00012480
      AJ(2,1) = -AJ(2,1) * DEI                                          00012490
      AJ(2,2) =  DUM                                                    00012500
C                                                                       00012510
      CR = 0.DO                                                         00012520
      DO  21  I = 1 , NNE                                               00012530
      M1 = NN(L,I)                                                      00012540
   21 CR = CR + SHP(I) * R(M1)                                          00012550
C                                                                       00012560
CCC                                                                     00012570
CCC      EVALUATION OF THE UPPER TRIANGLE OF  CE.                       00012580
CCC                                                                     00012590
C                                                                       00012600
      DUM = FACTC * DET * CR * GC(N) * GC(M)                            00012610
      DO  23  I = 1 , NNE                                               00012620
      DO  23  J = I , NNE                                               00012630
      CE(I,J) = CE(I,J) + DUM * SHP(I) * SHP(J)                         00012640
   23 CONTINUE                                                          00012650
C                                                                       00012660
CCC                                                                     00012670
CCC      CONSTRUCTION OF DNZR :  THE MATRIX OF THE PARTIAL              00012680
CCC      DERIVATIVES OF SHAPE FUNCTIONS RESPECT TO Z AND R              00012690
CCC                                                                     00012700
C                                                                       00012710
      DO  30  I = 1 , NNE                                               00012720
      DNZR(1,I) = AJ(1,1) * DXI(I) + AJ(1,2) * DETA(I)                  00012730
      DNZR(2,I) = AJ(2,1) * DXI(I) + AJ(2,2) * DETA(I)                  00012740
   30 CONTINUE                                                          00012750
C                                                                       00012760
CCC                                                                     00012770
CCC      EVALUATION OF UPPER TRIANGLE OF YE.                            00012780
CCC                                                                     00012790
C                                                                       00012800
      DUM = FACTY * CR * DET * GC(N) * GC(M)                            00012810
      DO  26  I = 1 , NNE                                               00012820
      DO  26  J = I , NNE                                               00012830
      DUM1 = DNZR(1,I) *DNZR(1,J)                                       00012840
      DUM2 = DNZR(2,I) * DNZR(2,J)                                      00012850
      YE(I,J) = YE(I,J) + (DUM1+DUM2) * DUM                             00012860
   26 CONTINUE                                                          00012870
C                                                                       00012880
C                                                                       00012890
    9 CONTINUE                                                          00012900
C                                                                       00012910
```

```
C                                                                       00012920
      DO  27  I = 1 , NNE                                               00012930
      DO  27  J = 1 , NNE                                               00012940
      CE(J,I) = CE(I,J)                                                 00012950
   27 YE(J,I) = YE(I,J)                                                 00012960
C                                                                       00012970
CC                                                                      00012980
CC       GENERATING THE THERMAL CAPACITANCE AND ADMITTANCE              00012990
CC       MATRICES IN THE BANDED FORM.                                   00013000
CC                                                                      00013010
C                                                                       00013020
      DO  14  I = 1 , NNE                                               00013030
      II = NN(L,I) - 1                                                  00013040
      DO  14  J = 1 , NNE                                               00013050
      J1 = NN(L,J) - 1                                                  00013060
      IF  (II.GT.J1)  GO  TO  14                                        00013070
      J1 = J1 - II                                                      00013080
      JJ = II+I                                                         00013090
      KK = J1 + 1                                                       00013100
      C(JJ,KK) = C(JJ,KK) + CE(I,J)                                     00013110
      Y(JJ,KK) = Y(JJ,KK) + YE(I,J)                                     00013120
   14 CONTINUE                                                          00013130
C                                                                       00013140
C                                                                       00013150
   25 CONTINUE                                                          00013160
C                                                                       00013170
C                                                                       00013180
      IF  (Q2.GT.0.D0)  GO  TO  40                                      00013190
      NC = 0                                                            00013200
      GO  TO  50                                                        00013210
   40 NC = NCFOT                                                        00013220
      DO  64  I = 1 , NCFOT                                             00013230
      DIAG(I) = 0.D0                                                    00013240
      OFF1(I) = 0.D0                                                    00013250
   64 OFF2(I) = 0.D0                                                    00013260
C                                                                       00013270
C                                                                       00013280
      NECFO = (NCFOT-1) / 2                                             00013290
      DO  60  M = 1 , NECFO                                             00013300
      K = 2 * M -1                                                      00013310
C                                                                       00013320
      JL = NCFO(K  )                                                    00013330
      JM = NCFO(K+1)                                                    00013340
      JR = NCFO(K+2)                                                    00013350
C                                                                       00013360
      DO  60  N = 1 , NGP                                               00013370
C                                                                       00013380
      XI = GX(N)                                                        00013390
```

```
C                                                                       00013400
      SHP1    = -.5D0 * (1.D0-XI) * XI                                  00013410
      SHP2    =   1.D0 - XI**2                                          00013420
      SHP3    =  .5D0 * (1.D0+XI) * XI                                  00013430
C                                                                       00013440
      DXI1    = -.5D0 + XI                                              00013450
      DXI2    = -2.D0 * XI                                              00013460
      DXI3    =  .5D0 + XI                                              00013470
C                                                                       00013480
      RR = SHP3*R(JL) + SHP2*R(JM) + SHP1*R(JR)                         00013490
C                                                                       00013500
      DRXI = DXI3*R(JL) + DXI2*R(JM) + DXI1*R(JR)                       00013510
      DZXI = DXI3*Z(JL) + DXI2*Z(JM) + DXI1*Z(JR)                       00013520
C                                                                       00013530
      DRXI = DRXI**2                                                    00013540
      DZXI = DZXI**2                                                    00013550
C                                                                       00013560
      DUMY = RR * GC(N) * DSQRT(DRXI+DZXI)                              00013570
C                                                                       00013580
      DIAG( K ) = DIAG( K ) + SHP3*SHP3 * DUMY                          00013590
      DIAG(K+1) = DIAG(K+1) + SHP2*SHP2 * DUMY                          00013600
      DIAG(K+2) = DIAG(K+2) + SHP1*SHP1 * DUMY                          00013610
      OFF1( K ) = OFF1( K ) + SHP3*SHP2 * DUMY                          00013620
      OFF1(K+1) = OFF1(K+1) + SHP2*SHP1 * DUMY                          00013630
      OFF2( K ) = OFF2( K ) + SHP1*SHP3 * DUMY                          00013640
C                                                                       00013650
C                                                                       00013660
   60 CONTINUE                                                          00013670
C                                                                       00013680
C                                                                       00013690
      CONST = 2.D0 * PAI * HTCO                                         00013700
C                                                                       00013710
      DO  67  I = 1 , NCFOT                                             00013720
      DIAG(I) = DIAG(I) * CONST                                         00013730
      OFF1(I) = OFF1(I) * CONST                                         00013740
   67 OFF2(I) = OFF2(I) * CONST                                         00013750
C                                                                       00013760
C                                                                       00013770
C        ADDING THE CONTRIBUTION OF Y STAR TO Y MATRIX                  00013780
C        FOR  OUTSIDE CONVECTION                                        00013790
C                                                                       00013800
C                                                                       00013810
      DO  68  I = 1 , NCFOT                                             00013820
      K = NCFC(I)                                                       00013830
   68 Y(K,1) = Y(K,1) + DIAG(I)                                         00013840
C                                                                       00013850
      N1 = NCFOT - 1                                                    00013860
      DO  69  I = 1 , N1                                                00013870
```

```
      K1 = NCFO(I)                                                      00013880
      K2 = NCFO(I+1)                                                    00013890
      K = K2 - K1 + 1                                                   00013900
      IF  (K.LE.0)   K = - K + 2                                        00013910
      K1 = MINO(K1,K2)                                                  00013920
      Y(K1,K) = Y(K1,K) + OFF1(I)                                       00013930
   69 CONTINUE                                                          00013940
C                                                                       00013950
      N2 = NCFOT-2                                                      00013960
      DO  66  I = 1 , N2 , 2                                            00013970
      K1 = NCFO(I)                                                      00013980
      K2 = NCFO(I+2)                                                    00013990
      K = K2 - K1 + 1                                                   00014000
      IF  (K.LE.0)   K = - K + 2                                        00014010
      K1 = MINO(K1,K2)                                                  00014020
      Y(K1,K) = Y(K1,K) + OFF2(I)                                       00014030
   66 CONTINUE                                                          00014040
C                                                                       00014050
C                                                                       00014060
C                                                                       00014070
   50 IF  (Q3.GT.0.D0)  GO  TO  41                                      00014080
      GO  TO  83                                                        00014090
   41 DO  54  I = 1 , NCFIT                                             00014100
      J = NC + I                                                        00014110
      DIAG(J) = 0.D0                                                    00014120
      OFF1(J) = 0.D0                                                    00014130
   54 OFF2(J) = 0.D0                                                    00014140
C                                                                       00014150
C                                                                       00014160
      NECFI = (NCFIT - 1) / 2                                           00014170
      DO  59  M = 1 , NECFI                                             00014180
C                                                                       00014190
      K = 2 * M -1                                                      00014200
C                                                                       00014210
      JL = NCFI(K)                                                      00014220
      JM = NCFI(K+1)                                                    00014230
      JR = NCFI(K+2)                                                    00014240
C                                                                       00014250
      DO  59  N = 1 , NGP                                               00014260
C                                                                       00014270
      XI = GX(N)                                                        00014280
C                                                                       00014290
C                                                                       00014300
      SHP1   = -.5D0 * (1.D0 - XI) * XI                                 00014310
      SHP2   =  1.D0 - XI**2                                            00014320
      SHP3   =  .5D0 * (1.D0 + XI) * XI                                 00014330
      DXI1   = -.5D0 + XI                                               00014340
      DXI2   = -2.D0 * XI                                               00014350
```

```
      DXI3    =   .5D0 + XI                                                  00014360
                                                                             00014370
C                                                                            00014380
      RR = SHP1*R(JL) + SHP2*R(JM) + SHP3*R(JR)                              00014390
C                                                                            00014400
      DRXI = DXI1*R(JL) + DXI2*R(JM) + DXI3*R(JR)                            00014410
      DZXI = DXI1*Z(JL) + DXI2*Z(JM) + DXI3*Z(JR)                            00014420
C                                                                            00014430
      DRXI = DRXI**2                                                         00014440
      DZXI = DZXI**2                                                         00014450
C                                                                            00014460
      DUMY = RR * GC(N) * DSQRT(DRXI+DZXI)                                   00014470
C                                                                            00014480
      I = NC + K                                                             00014490
C                                                                            00014500
      DIAG( I ) = DIAG( I )+ SHP1*SHP1 * DUMY                                00014510
      DIAG(I+1) = DIAG(I+1)+ SHP2*SHP2 * DUMY                                00014520
      DIAG(I+2) = DIAG(I+2)+ SHP3*SHP3 * DUMY                                00014530
      OFF1( I ) = OFF1( I ) + SHP1*SHP2 * DUMY                               00014540
      OFF1(I+1) = OFF1(I+1) + SHP2*SHP3 * DUMY                               00014550
      OFF2( I ) = OFF2( I ) + SHP1*SHP3 * DUMY                               00014560
CC                                                                           00014570
   59 CONTINUE                                                               00014580
CC                                                                           00014590
CC                                                                           00014600
      CONST = 2.D0 * PAI * HTCI                                              00014610
C                                                                            00014620
      DO  57  J = 1 , NCFIT                                                  00014630
      I = NC + J                                                             00014640
      DIAG(I) = DIAG(I) * CONST                                              00014650
      OFF1(I) = OFF1(I) * CONST                                              00014660
   57 OFF2(I) = OFF2(I) * CONST                                              00014670
C                                                                            00014680
CCCCC                                                                        00014690
CCCCC    ADDING THE CONTRIBUTION OF Y STAR TO Y MATRIX                       00014700
CCCCC    FOR INSIDE CONVECTION                                               00014710
CCCCC                                                                        00014720
C                                                                            00014730
      DO  58  I = 1 , NCFIT                                                  00014740
      J = NC + I                                                             00014750
      K = NCFI(I)                                                            00014760
   58 Y(K,1) = Y(K,1) + DIAG(J)                                              00014770
      M1 = NCFIT - 1                                                         00014780
C                                                                            00014790
      DO  55  I = 1 , M1                                                     00014800
      J = NC + I                                                             00014810
      K1 = NCFI(I)                                                           00014820
      K2 = NCFI(I+1)                                                         00014830
```

```
      K = K2 - K1 + 1                                                   00014840
      IF  (K.LE.0)   K = - K + 2                                        00014850
      K1 = MINO(K1,K2)                                                  00014860
      Y(K1,K) = Y(K1,K) + OFF1(J)                                       00014870
   55 CONTINUE                                                          00014880
C                                                                       00014890
      M2 = NCFIT - 2                                                    00014900
      DO  56  I = 1 , M2 , 2                                            00014910
      J = NC + I                                                        00014920
      K1 = NCFI(I)                                                      00014930
      K2 = NCFI(I+2)                                                    00014940
      K = K2 - K1 + 1                                                   00014950
      IF  (K.LE.0)   K = - K + 2                                        00014960
      K1 = MINO(K1,K2)                                                  00014970
      Y(K1,K) = Y(K1,K) + OFF2(J)                                       00014980
   56 CONTINUE                                                          00014990
C                                                                       00015000
   83 CONTINUE                                                          00015010
CC                                                                      00015020
CCC                                                                     00015030
CCCC                                                                    00015040
CCCCC       EVALUATION OF A AND G MATRICES                              00015050
CCCC                                                                    00015060
CCC                                                                     00015070
      DUM = .5000 * DTI                                                 00015080
      DO  1  I=1,NNT                                                    00015090
      DO  1  J = 1 , NBAND                                              00015100
      EL = DUM * Y(I,J)                                                 00015110
      C1 = C(I,J) + EL                                                  00015120
      C2 = C(I,J) - EL                                                  00015130
      C(I,J) = C1                                                       00015140
      Y(I,J) = C2                                                       00015150
    1 CONTINUE                                                          00015160
C                                                                       00015170
CCC                                                                     00015180
CCCC    C  NOW CONTAINS  C+.5*DTI*Y  AND  Y CONTAINS  C-.5*DTI*Y        00015190
CCCCC                                                                   00015200
CCCCCC                                                                  00015210
CCCCCC                                                                  00015220
CCCCCC                                                                  00015230
CCCCC       THIS PART WILL DO THE CHOLESKY DECOMPOSITION OF             00015240
CCCC        THE MATRIX <C> WHICH IS ALREADY PUT IN THE BANDED           00015250
CCC         FORM. ORIGINAL VERSION WAS CODED BY PROF. CAN.IN            00015260
CC                                                                      00015270
C                                                                       00015280
      DO  300  I = 1 , NNT                                              00015290
      DUM = C(I,1)                                                      00015300
      DUM = DSQRT(DUM)                                                  00015310
```

```
      DO  200  J = 1 , NBAND                                          00015320
  200 C(I,J) = C(I,J) / DUM                                           00015330
C                                                                     00015340
      DO  260  J = 2 , NBAND                                          00015350
      L = I + J - 1                                                   00015360
      IF  (L.GT.NNT)  GO  TO  260                                     00015370
      AA = C(I,J)                                                     00015380
      IF  (AA.EQ.0.D0)  GO  TO  260                                   00015390
      DO  250  K = J , NBAND                                          00015400
      M = K - J + 1                                                   00015410
  250 C(L,M) = C(L,M) - AA * C(I,K)                                   00015420
  260 CONTINUE                                                        00015430
  300 CONTINUE                                                        00015440
C                                                                     00015450
C                                                                     00015460
      RETURN                                                          00015470
      END                                                             00015480
```

```
C                                                                       00015490
      SUBROUTINE  FLOW                                                  00015500
C                                                                       00015510
CCCCCCCCC                                                               00015520
     ************************************************************       00015530
     *                                                          *       00015540
     *  FOR A GIVEN OUTSIDE OR INSIDE ENTRY TIME VARIATION OF   *       00015550
     *  FLUID TEMPERATURE , THE SUBROUTINE FLOW WILL EVALUATE   *       00015560
     *  THE TEMPERATURE OF THE FLUID NODES AT A GIVEN TIME .    *       00015570
     *  THE CALCULATION IS BASED UPON CONSTANT FLUX.            *       00015580
     *                                                          *       00015590
     ************************************************************       00015600
C                                                                       00015610
      IMPLICIT  REAL * 8 (A-H,O-Z)                                      00015620
      REAL   * 4  ENTEO,ENTEI,ENTIO,ENTII                               00015630
C                                                                       00015640
C                                                                       00015650
      COMMON /ONE/ AJ(2,2),B(4,16),COPY(37,298),D(4,4),DL(4,4),         00015660
     1DNZR(2,8),ESM(16,16),F(24,298),SSM(298,66),STOR(20,150)           00015670
C                                                                       00015680
      COMMON /TWO/ AL(5),DENS(5),DETA(8),DXI(8),E(5),EPSO(4),           00015690
     1FE(16),GC(4),GX(4),PDI(5),R(149),SHP(8),SHT(5),TK(5),Z(149)       00015700
C                                                                       00015710
      COMMON /TRE/ AXIALF,DTI,EVERY,FORS1,OMEGA,HTCI,HTCO,              00015720
     1POSI,PRES,Q1,Q2,Q3,SHMAX,SMAX,TIME1,TINIT,TMAXM,TREF,             00015730
     2TMAXO,RMAXM,RMAXO,ZMAXM,ZMAXO,Q4,BIG                              00015740
C                                                                       00015750
      COMMON /FOR/ NN(40,9)                                             00015760
C                                                                       00015770
      COMMON /FIV/ NCFI(37),NCFO(37),NNR(37),NRE(37),NNL(37),           00015780
     1NPN(37)                                                           00015790
      COMMON /SIX/ IBAND,IEXT,INTP,IVEC,JVEC,KK,LOAD,NCFIT,NCFOT,       00015800
     1NCHECK,NET,NDF,NGP,NNE,NNLT,NNRT,NPASS,NPROB,NRPPI,NRPPO,         00015810
     3NNT,NNRE,NPNT,IPLANE,NBAND                                        00015820
C                                                                       00015830
      EQUIVALENCE  (C,SSM) , (Y,SSM(1,18)) , (T,SSM(1,35)) ,            00015840
     1(BTE,SSM(1,36)) , (TEMP,SSM(1,37)) , (V,SSM(1,38)) ,              00015850
     2(FT1,SSM(1,39)) , (FT2,SSM(1,40)) , (CE,SSM(1,41)) ,              00015860
     3(YE,SSM(1,42)) , (OFF1,SSM(1,43)) , (OFF2,SSM(1,44)) ,            00015870
     4(DIAG,SSM(1,45)) , (BDRYI,SSM(1,46)) , (BDRYO,SSM(1,47)) ,        00015880
     5(ENTII,SSM(1,48)) , (ENTIO,SSM(1,49)), (ENTEI,SSM(1,50)),         00015890
     6(ENTEO,SSM(1,51)),(VOL,SSM(1,52)),(VOLO,SSM(1,53))                00015900
C                                                                       00015910
      DIMENSION  C(149,33),Y(149,33),T(149),BTE(149),TEMP(149),         00015920
     1V(149),FT1(90),FT2(90),CE(8,8),YE(8,8),OFF1(37),OFF2(37),         00015930
                                                                        00015940
```

```
     2DIAG(37),BDRYI(16,5),BDRYO(16,5),VOL(36),VOLO(36),ENTEO(75),      00015950
     3ENTEI(75),ENTIO(75),ENTII(75)                                     00015960
C                                                                       00015970
CC                                                                      00015980
      C1 = .81157810217736320                                           00015990
      C2 = 1.6755160819145570                                           00016000
      C3 = .0261799387799149 50                                         00016010
      C4 = 1.2042771838760882                                           00016020
      C5 = -.2094395102393196                                           00016030
      C6 = -.3665191429188094                                           00016040
C                                                                       00016050
      PAI = 3.141592653589793                                           00016060
C                                                                       00016070
CCC      FLOW IN THE LONGITUDINAL DIRECTION , AS IN PIPE                00016080
CCC                                                                     00016090
C                                                                       00016100
      N = 0                                                             00016110
      IF  (Q2.EQ.0.D0)  GO  TO  46                                      00016120
      N = NCFOT                                                         00016130
      IF  (KK.EQ. 1)    GO  TO  40                                      00016140
      IF  (NCHECK.EQ.0) GO  TO  45                                      00016150
C                                                                       00016160
CC                                                                      00016170
      DO  41  I = 1 , NCFOT                                             00016180
   41 FT1(I) = FT2(I)                                                   00016190
C                                                                       00016200
   40 TIME = KK * DTI                                                   00016210
CC                                                                      00016220
CC                                                                      00016230
      M = KK / 2                                                        00016240
      I = KK - 2 * M                                                    00016250
      M = M + 1                                                         00016260
      IF ((KK.LE.150).AND.(I.EQ.1))  ENTEO(M) = FT1(1)                  00016270
      IF ((KK.LE.150).AND.(I.EQ.1))  ENTIO(M) = TIME-DTI                00016280
C                                                                       00016290
CC                                                                      00016300
      DO  42  I = 2 , NRPPO                                             00016310
   42 IF  (TIME.GT.BDRYO(I,1))  VEL = BDRYO(I,5)                        00016320
C                                                                       00016330
      IF  (VEL.EQ.0.D0)  GO  TO  46                                     00016340
C                                                                       00016350
CC                                                                      00016360
      IF  (KK.GT.1)  GO  TO  75                                         00016370
      NECFO = (NCFOT-1) / 2                                             00016380
C                                                                       00016390
CC                                                                      00016400
      DUM = 0.D0                                                        00016410
      DO  10  I=2,NCFOT                                                 00016420
```

```
      R1 = R(NCFO(I-1))                                                 00016430
      R2 = R(NCFO(I))                                                   00016440
      IF  (R1.GE.R2)  DUM = R1                                          00016450
   10 CONTINUE                                                          00016460
      IF  (DUM.EQ.0.D0)  DUM = R(NCFO(NCFOT))                           00016470
C                                                                       00016480
C                                                                       00016490
C                                                                       00016500
      DUM = DUM + 1.D0                                                  00016510
      AREAO = PAI * DUM**2                                              00016520
      VOLO(1) = 0.D0                                                    00016530
      DO  70  I = 1 , NECFO                                             00016540
      K = 2 * I - 1                                                     00016550
      J1 = NCFO( K )                                                    00016560
      J2 = NCFO(K+1)                                                    00016570
      J3 = NCFO(K+2)                                                    00016580
C                                                                       00016590
      DUM = C1*R(J1)**2+C2*R(J2)**2+C3*R(J3)**2                         00016600
      DUM=DUM+C4*R(J1)*R(J2)+C5*R(J1)*R(J3)+C6*R(J2)*R(J3)              00016610
      VOLO(K+1) = (AREAO-DUM)*DABS(Z(J2)-Z(J1))                         00016620
C                                                                       00016630
C                                                                       00016640
      DUM = C3*R(J1)**2+C2*R(J2)**2+C1*R(J3)**2                         00016650
      DUM=DUM+C6*R(J1)*R(J2)+C5*R(J1)*R(J3)+C4*R(J2)*R(J3)              00016660
      VOLO(K+2) = (AREAO-DUM)*DABS(Z(J3)-Z(J2))                         00016670
   70 CONTINUE                                                          00016680
C                                                                       00016690
C                                                                       00016700
   75 DO  73  I = 2 , NRPPO                                             00016710
      IF  (TIME.GT.BDRYO(I,1))  KK2 = I                                 00016720
   73 CONTINUE                                                          00016730
      TIM = TIME                                                        00016740
      TF = BDRYO(KK2,4) * (TIM-BDRYO(KK2,1))                           00016750
      FT2(1) = BDRYO(KK2,2) + TF                                        00016760
C                                                                       00016770
C                                                                       00016780
      DUMY = VEL*(AREAO-PAI*R(NCFO(1))**2)                              00016790
C                                                                       00016800
C                                                                       00016810
      DO  74  I = 2 , NCFOT                                             00016820
      TIM1 = VOLO(I) / DUMY                                             00016830
      TIM = TIM - TIM1                                                  00016840
      DO  76  J = 2 , NRPPO                                             00016850
      IF  (TIM.GE.BDRYO(J,1))  KK2 = J                                  00016860
   76 CONTINUE                                                          00016870
C                                                                       00016880
C                                                                       00016890
      TF = BDRYO(KK2,4) * (TIM-BDRYO(KK2,1))                           00016900
```

```
      FT2(I) = BDRYO(KK2,2) + TF                                                00016910
   74 CONTINUE                                                                  00016920
      GO  TO  46                                                                00016930
   45 TIME = KK * DTI                                                           00016940
C                                                                               00016950
   46  IF  (Q3.EQ.0.D0)   GO  TO  56                                            00016960
C                                                                               00016970
C                                                                               00016980
      IF  (KK.EQ.1)      GO  TO  50                                            00016990
      IF(NCHECK.EQ.0)  GO  TO  55                                               00017000
C                                                                               00017010
      DO  51  I = 1 , NCFIT                                                     00017020
      J = N + I                                                                 00017030
   51 FT1(J) = FT2(J)                                                           00017040
C                                                                               00017050
   50 TIME = KK * DTI                                                           00017060
C                                                                               00017070
C                                                                               00017080
      M = KK / 2                                                                00017090
      I = KK - 2 * M                                                            00017100
      M = M + 1                                                                 00017110
      IF ((KK.LE.150).AND.(I.EQ.1))  ENTEI(M) = FT1(N+1)                        00017120
      IF ((KK.LE.150).AND.(I.EQ.1))  ENTII(M) = TIME-DTI                        00017130
C                                                                               00017140
      DO  52  I = 2 , NRPPI                                                     00017150
   52 IF  (TIME.GT.BDRYI(I,1))   VEL = BDRYI(I,5)                               00017160
C                                                                               00017170
      IF  (VEL.EQ.0.D0)   GO  TO  56                                            00017180
C                                                                               00017190
      IF  (KK.GT. 1)  GO  TO  82                                                00017200
      NECFI = (NCFIT-1)/2                                                       00017210
      AREA = PAI * R(NCFI(1))**2                                                00017220
      VOL(1) = 0.D0                                                             00017230
C                                                                               00017240
C                                                                               00017250
C                                                                               00017260
      DO  81  I = 1 , NECFI                                                     00017270
      K = 2 * I - 1                                                             00017280
      J1 = NCFI( K )                                                            00017290
      J2 = NCFI(K+1)                                                            00017300
      J3 = NCFI(K+2)                                                            00017310
C                                                                               00017320
      DUM = C1*R(J1)**2+C2*R(J2)**2+C3*R(J3)**2                                 00017330
      DUM=DUM+C4*R(J1)*R(J2)+C5*R(J1)*R(J3)+C6*R(J2)*R(J3)                      00017340
      VOL(K+1) = DUM * DABS(Z(J2)-Z(J1))                                        00017350
C                                                                               00017360
C                                                                               00017370
      DUM = C3*R(J1)**2+C2*R(J2)**2+C1*R(J3)**2                                 00017380
```

```
      DUM=DUM+C6*R(J1)*R(J2)+C5*R(J1)*R(J3)+C4*R(J2)*R(J3)                00017390
      VOL(K+2) = DUM * DABS(Z(J3) - Z(J2))                               00017400
   81 CONTINUE                                                           00017410
C                                                                        00017420
CC                                                                       00017430
CCC                                                                      00017440
CC                                                                       00017450
C                                                                        00017460
   82 DO  83  I = 2 , NRPPI                                              00017470
      IF  (TIME. GT .BDRYI(I,1))  KK2 = I                                00017480
   83 CONTINUE                                                           00017490
C                                                                        00017500
C                                                                        00017510
      TIM = TIME                                                         00017520
      TF = BDRYI(KK2,4) * (TIM-BDRYI(KK2,1))                             00017530
      FT2(N+1) = BDRYI(KK2,2) + TF                                       00017540
C                                                                        00017550
C                                                                        00017560
      DUMY = VEL * AREA                                                  00017570
C                                                                        00017580
C                                                                        00017590
      DO  84  I = 2 , NCFIT                                              00017600
      TIM1 = VOL(I) / DUMY                                               00017610
      TIM = TIM - TIM1                                                   00017620
      DO  85  J = 2 , NRPPI                                              00017630
      IF  (TIM.GE.BDRYI(J,1))    KK2 = J                                 00017640
   85 CONTINUE                                                           00017650
      TF = BDRYI(KK2,4) * (TIM-BDRYI(KK2,1))                             00017660
      FT2(N+I) = BDRYI(KK2,2) + TF                                       00017670
   84 CONTINUE                                                           00017680
      GO  TO  56                                                         00017690
C                                                                        00017700
C                                                                        00017710
   55 TIME = KK * DTI                                                    00017720
   56 RETURN                                                             00017730
      END                                                                00017740
```

```
C                                                                       00017750

      SUBROUTINE  FORMV                                                 00017760
C                                                                       00017770
CCC                                                                     00017780
CCC   ***************************************************************** 00017790
CCC                                                                   * 00017800
CCC   THE VECTOR V OF THE RIGHT HAND SIDE OF THE DISCRETIZED          * 00017810
CCC   FINITE ELEMENT FORMULATION FOR CONVECTION OR  CONSTANT          * 00017820
CCC   TEMPERATURE (OUTSIDE OR INSIDE) BOUNDARY CONDITION  IS          * 00017830
CCC   FORMED HERE.                                                    * 00017840
CCC                                                                   * 00017850
CCC   ***************************************************************** 00017860
CCC                                                                     00017870
C                                                                       00017880
      IMPLICIT  REAL * 8 (A-H,O-Z)                                      00017890
C                                                                       00017900
      COMMON /ONE/ AJ(2,2),B(4,16),COPY(37,298),D(4,4),DL(4,4),         00017910
     1DNZR(2,8),ESM(16,16),F(24,298),SSM(298,66),STOR(20,150)           00017920
C                                                                       00017930
      COMMON /TWO/ AL(5),DENS(5),DETA(8),DXI(8),E(5),EPSO(4),           00017940
     1FE(16),GC(4),GX(4),PCI(5),R(149),SHP(8),SHT(5),TK(5),Z(149)       00017950
C                                                                       00017960
      COMMON /TRE/ AXIALF,DTI,EVERY,FORS1,OMEGA,HTCI,HTCO,              00017970
     1POSI,PRES,Q1,Q2,Q3,SHMAX,SMAX,TIME,TIMEI,TINIT,TMAXM,TREF,        00017980
     2TMAXO,RMAXM,RMAXO,ZMAXM,ZMAXO,Q4,BIG                              00017990
C                                                                       00018000
      COMMON /FOR/ NN(40,9)                                             00018010
C                                                                       00018020
      COMMON /FIV/ NCFI(37),NCFO(37),NNR(37),NRE(37),NNL(37),           00018030
     1NPN(37)                                                           00018040
      COMMON /SIX/ IBAND,IEXT,INTP,IVEC,JVEC,KK,LOAD,NCFIT,NCFOT,       00018050
     1NCHECK,NET,NDF,NGP,NNE,NNLT,NNRT,NPASS,NPROB,NRPPI,NRPPO,         00018060
     3NNT,NNRE,NPNT,IPLANE,NBAND                                        00018070
C                                                                       00018080
      EQUIVALENCE  (C,SSM) , (Y,SSM(1,18)) , (T,SSM(1,35)) ,            00018090
     1(BTE,SSM(1,36)) , (TEMP,SSM(1,37)) , (V,SSM(1,38)) ,             00018100
     2(FT1,SSM(1,39)) , (FT2,SSM(1,40)) , (CE,SSM(1,41)) ,             00018110
     3(YE,SSM(1,42)) , (OFF1,SSM(1,43)) , (OFF2,SSM(1,44)) ,           00018120
     4(DIAG,SSM(1,45)) , (BDRYI,SSM(1,46)) , (BDRYO,SSM(1,47)) ,       00018130
     5(ENTII,SSM(1,48)) , (ENTIO,SSM(1,49)), (ENTEI,SSM(1,50)),        00018140
     6(ENTEO,SSM(1,51)),(VOL,SSM(1,52)),(VOLO,SSM(1,53))               00018150
C                                                                       00018160
      DIMENSION  C(149,33),Y(149,33),T(149),BTE(149),TEMP(149),         00018170
     1V(149),FT1(90),FT2(90),CE(8,8),YE(8,8),OFF1(37),OFF2(37),        00018180
     2DIAG(37),BDRYI(16,5),BDRYO(16,5),VOL(36),VOLO(36),ENTEO(75),     00018190
     3ENTEI(75),ENTIO(75),ENTII(75)                                     00018200
```

```
C                                                                       00018210
C                                                                       00018220
C                                                                       00018230
      N = 0                                                             00018240
      IF  (Q2.GT.0.D0)   GO  TO  5                                      00018250
      GO  TO  6                                                         00018260
    5 N = NCFOT                                                         00018270
      DO  1  I = 1 , NCFOT                                              00018280
      J = NCFO(I)                                                       00018290
    1 V(J) = 0.D0                                                       00018300
C                                                                       00018310
      DO  2  J = 1 , NCFOT                                              00018320
      I = NCFO(J)                                                       00018330
    2 V(I) = V(I) + DIAG(J) * (FT1(J)+FT2(J))                           00018340
C                                                                       00018350
      N1 = NCFOT - 1                                                    00018360
      DO  3  I = 1 , N1                                                 00018370
      I1 = I +1                                                         00018380
      K1 = NCFO( I )                                                    00018390
      K2 = NCFO(I+1)                                                    00018400
      V(K1) = V(K1) + OFF1(I) * (FT1(I1)+FT2(I1))                       00018410
    3 V(K2) = V(K2) + OFF1(I) * (FT1(I )+FT2(I ))                       00018420
C                                                                       00018430
      N2 = NCFOT - 2                                                    00018440
      DO  4  I =  1 , N2 , 2                                            00018450
      I2 = I + 2                                                        00018460
      K2 = NCFO(I+2)                                                    00018470
      J = NCFO(I)                                                       00018480
      V(J ) = V(J ) + OFF2(I) * (FT1(I2)+FT2(I2))                       00018490
    4 V(K2) = V(K2) + OFF2(I) * (FT1(I)+FT2(I))                         00018500
C                                                                       00018510
C                                                                       00018520
    6 IF  (Q3.GT.0.D0)  GO  TO  11                                      00018530
      GO  TO  12                                                        00018540
   11 DO  7  I = 1 , NCFIT                                              00018550
      J = NCFI(I)                                                       00018560
    7 V(J) = 0.D0                                                       00018570
C                                                                       00018580
      DO  8  J = 1 , NCFIT                                              00018590
      I = NCFI(J)                                                       00018600
      K = N + J                                                         00018610
    8 V(I) = V(I) + DIAG(K) * (FT1(K)+FT2(K))                           00018620
C                                                                       00018630
      N1 = NCFIT - 1                                                    00018640
      DO  9  I = 1 , N1                                                 00018650
      K = N + I                                                         00018660
      I1 = K + 1                                                        00018670
      K1 = NCFI( I )                                                    00018680
      K2 = NCFI(I+1)
```

```
      V(K1) = V(K1) + OFF1(K) * (FT1(I1)+FT2(I1))                       00018690
    9 V(K2) = V(K2) + OFF1(K) * (FT1(K )+FT2(K ))                       00018700
C                                                                       00018710
      N2 = NCFIT - 2                                                    00018720
      DO  10  I = 1 , N2 , 2                                            00018730
      K = N + I                                                         00018740
      I2 = K + 2                                                        00018750
      J = NCFI( I )                                                     00018760
      K2 = NCFI(I+2)                                                    00018770
      V(J ) = V(J ) + OFF2(K) * (FT1(I2)+FT2(I2))                       00018780
   10 V(K2) = V(K2) + OFF2(K) * (FT1(K)+FT2(K))                         00018790
C                                                                       00018800
   12 CONTINUE                                                          00018810
      RETURN                                                            00018820
      END                                                               00018830
```

```
      SUBROUTINE  TEMPER                                                    00018840
C                                                                           00018850
CC                                                                          00018860
CC    ***************************************************************     00018870
CC                                                                  *     00018880
CC    IN THIS SUBROUTINE THE NODAL TEMPERATURES  ARE                *     00018890
CC    EVALUATED BY TRAPEZOIDAL INTEGRATION. ALSO FOR                *     00018900
CC    EVERY 10 STEPS OF TIME INTEGRATION THE  'IRONS                *     00018910
CC    CORRECTION'  IS APPLIED .                                     *     00018920
CC                                                                  *     00018930
CC    ***************************************************************     00018940
CC                                                                          00018950
C                                                                           00018960
      IMPLICIT  REAL * 8 (A-H,O-Z)                                          00018970
C                                                                           00018980
      COMMON /ONE/ AJ(2,2),B(4,16),COPY(37,298),D(4,4),DL(4,4),             00018990
     1DNZR(2,8),ESM(16,16),F(24,298),SSM(298,66),STOR(20,150)               00019000
C                                                                           00019010
      COMMON /TWO/ AL(5),DENS(5),DETA(8),DXI(8),E(5),EPSO(4),               00019020
     1FE(16),GC(4),GX(4),POI(5),R(149),SHP(8),SHT(5),TK(5),Z(149)           00019030
C                                                                           00019040
      COMMON /TRE/ AXIALF,DTI,EVERY,FORS1,OMEGA,HTCI,HTCO,                  00019050
     1POSI,PRES,Q1,Q2,Q3,SHMAX,SMAX,TIME,TIME1,TINIT,TMAXM,TREF,            00019060
     2TMAXO,RMAXM,RMAXO,ZMAXM,ZMAXO,Q4,BIG                                  00019070
C                                                                           00019080
      COMMON /FOR/ NN(40,9)                                                 00019090
C                                                                           00019100
      COMMON /FIV/ NCFI(37),NCFO(37),NNR(37),NRE(37),NNL(37),               00019110
     1NPN(37)                                                               00019120
      COMMON /SIX/ IBAND,IEXT,INTP,IVEC,JVEC,KK,LOAD,NCFIT,NCFOT,           00019130
     1NCHECK,NET,NDF,NGP,NNE,NNLT,NNRT,NPASS,NPROB,NRPPI,NRPPO,             00019140
     3NNT,NNRE,NPNT,IPLANE,NBAND                                            00019150
C                                                                           00019160
      EQUIVALENCE  (C,SSM) , (Y,SSM(1,18)) , (T,SSM(1,35)) ,                00019170
     1(BTE,SSM(1,36)) , (TEMP,SSM(1,37)) , (V,SSM(1,38)) ,                  00019180
     2(FT1,SSM(1,39)) , (FT2,SSM(1,40)) , (CE,SSM(1,41)) ,                  00019190
     3(YE,SSM(1,42)) , (OFF1,SSM(1,43)) , (OFF2,SSM(1,44)) ,                00019200
     4(DIAG,SSM(1,45)) , (BDRYI,SSM(1,46)) , (BDRYO,SSM(1,47)) ,            00019210
     5(ENTII,SSM(1,48)) , (ENTIO,SSM(1,49)) , (ENTEI,SSM(1,50)),            00019220
     6(ENTEO,SSM(1,51)),(VOL,SSM(1,52)),(VOLO,SSM(1,53))                    00019230
C                                                                           00019240
      DIMENSION  C(149,33),Y(149,33),T(149),BTE(149),TEMP(149),             00019250
     1V(149),FT1(90),FT2(90),CE(8,8),YE(8,8),OFF1(37),OFF2(37),             00019260
     2DIAG(37),BDRYI(16,5),BDRYO(16,5),VOL(36),VOLO(36),ENTEO(75),          00019270
     3ENTEI(75),ENTIO(75),ENTII(75)                                         00019280
C                                                                           00019290
```

```
C                                                                       00019300
CC                                                                      00019310
      NCHECK = 1                                                        00019320
C                                                                       00019330
      DUM = DTI / 2.DO                                                  00019340
      DO  1  I=1,NNT                                                    00019350
    1 V(I) = DUM * V(I)                                                 00019360
C                                                                       00019370
CCCC                                                                    00019380
CCCC     THIS PART WILL MULTIPLY THE BANDED MATRIX Y BY THE             00019390
CCCC     COLUMN VECTOR OF TEMPERATURE AND ADD TO V.                     00019400
C                                                                       00019410
      DO  7  J = 1 , NNT                                                00019420
      BTE(J) = 0.DO                                                     00019430
      M = MIN0(NBAND,NNT+1-J)                                           00019440
      DO  5  K = 1 , M                                                  00019450
      L = J + K - 1                                                     00019460
      BTE(J) = BTE(J) + Y(J,K) * TEMP(L)                                00019470
    5 CONTINUE                                                          00019480
C                                                                       00019490
      IF  (J. LT .2)  GO  TO  7                                         00019500
      L = MIN0(J , NBAND)                                               00019510
      DO  6  K = 2 , L                                                  00019520
      M = J - K + 1                                                     00019530
      BTE(J) = BTE(J) + Y(M,K) * TEMP(M)                                00019540
    6 CONTINUE                                                          00019550
    7 BTE(J) = BTE(J) + V(J)                                            00019560
C                                                                       00019570
CCCC                                                                    00019580
CCCC     SOLVING FOR TEMPERATURE BY FORWARD AND BACK SUBSTITUTION       00019590
CCCC                                                                    00019600
C                                                                       00019610
C                                                                       00019620
      DO  32  I = 1 , NNT                                               00019630
      BTE(I) = BTE(I) / C(I,1)                                          00019640
      DO  31  J = 2 , NBAND                                             00019650
      L = I + J - 1                                                     00019660
      IF  (L. GT .NNT)  GO  TO  32                                      00019670
      DUM = C(I,J)                                                      00019680
   31 BTE(L) = BTE(L) - DUM * BTE(I)                                    00019690
   32 CONTINUE                                                          00019700
C                                                                       00019710
      BTE(NNT) = BTE(NNT) / C(NNT,1)                                    00019720
      DO  34  L = 2 , NNT                                               00019730
      K = NNT - L + 1                                                   00019740
      SUM = 0.DO                                                        00019750
      DO  33  J = 2 , NBAND                                             00019760
                                                                        00019770
```

```
      M = J + K - 1                                                     00019780
      IF  (NNT. LT .M)    GO  TO  34                                    00019790
   33 SUM = SUM + C(K,J) * BTE(M)                                       00019800
   34 BTE(K) = (BTE(K) -SUM) / C(K,1)                                   00019810
C                                                                       00019820
C                                                                       00019830
      IF  (KK.GT.2)  GO  TO  2                                          00019840
      KO = 0                                                            00019850
C                                                                       00019860
CC       APPLICATION OF IRONS CORRECTION.                               00019870
C                                                                       00019880
    2 K1 = KO + 12 - KK                                                 00019890
      GO  TO (70, 60 ,50) ,K1                                           00019900
      GO  TO  14                                                        00019910
C                                                                       00019920
   50 DO  51  I = 1 , NNT                                               00019930
   51 T(I) = .25D0 * BTE(I)                                             00019940
      GO  TO  14                                                        00019950
C                                                                       00019960
   60 DO  61  I = 1 , NNT                                               00019970
   61 T(I) = T(I) + .5D0 * BTE(I)                                       00019980
      GO  TO  14                                                        00019990
C                                                                       00020000
   70 DO  71  I = 1 , NNT                                               00020010
   71 TEMP(I) = T(I) + .25D0 * BTE(I)                                   00020020
C                                                                       00020030
CC       TEMPERATURE IS CORRECTED NOW                                   00020040
C                                                                       00020050
      NCHECK = 0                                                        00020060
      KO = KO + 10                                                      00020070
      KK = KK - 1                                                       00020080
      TIME = KK*DTI                                                     00020090
      IP = 0                                                            00020100
C                                                                       00020110
      GO  TO  16                                                        00020120
   14 DO  4  I = 1,NNT                                                  00020130
    4 TEMP(I) = BTE(I)                                                  00020140
      IP = 1                                                            00020150
C                                                                       00020160
   16 CONTINUE                                                          00020170
C                                                                       00020180
      C1 = 1000.D0 * TIME                                               00020190
      DO  21  J = 1 , IVEC                                              00020200
      C2 =(TIME1 + EVERY * (J-1)) * 1000.D0                             00020210
      DC=DABS(C1-C2)                                                    00020220
      IF  (DC.LE..0001)  GO  TO  20                                     00020230
   21 CONTINUE                                                          00020240
C                                                                       00020250
```

```
      GO  TO  22                                                        00020260
C                                                                       00020270
C                                                                       00020280
   20 DO  40  I = 1 , NNT                                               00020290
   40 STOR(J,I) = TEMP(I)                                               00020300
      STOR(J,NNT+1) = TIME                                              00020310
C                                                                       00020320
C                                                                       00020330
   22 IF  (INTP.EQ.0)  GO  TO  81                                       00020340
      INT1 = INTP                                                       00020350
   83 KUNT = KK / INT1                                                  00020360
      IF  (KUNT.EQ.0)  GO  TO  81                                       00020370
      KUNT = KK - KUNT * INT1                                           00020380
C                                                                       00020390
      IF  (KUNT.EQ.0)  GO  TO  82                                       00020400
      INT1 = INT1 + INTP                                                00020410
      GO  TO  83                                                        00020420
C                                                                       00020430
   82 ID = INTP / 10                                                    00020440
      ID = INTP - ID * 10                                               00020450
      IF  ((ID.EQ.0).AND.(IP.NE.0))  GO TO 81                           00020460
      WRITE  (6,101)  TIME                                              00020470
  101 FORMAT (/,5X,'AT TIME=',F8.2,' THE TEMPERATURES AT THE ',         00020480
     1'NODES ARE :',/)                                                  00020490
      WRITE  (6,222)  (I,TEMP(I),I=1,NNT)                               00020500
  222 FORMAT  (7(1X,'(',I3,')',1PD13.5))                                00020510
   81 KK = KK + 1                                                       00020520
C                                                                       00020530
C                                                                       00020540
      RETURN                                                            00020550
      END                                                               00020560
C                                                                       00020570
```

```
      SUBROUTINE  STIFF                                                 00020580
C                                                                       00020590
CCC                                                                     00020600
CCC      ***************************************************************00020610
CCC                                                                    *00020620
CCC      SUBROUTINE STIFF , CALCULATES THE STIFFNESS MATRIX            *00020630
CCC      FOR A GIVEN PROBLEM IN THE BANDED FORM,APPLIES THE            *00020640
CCC      REQUIRED BOUNDARY CONDITIONS AND AT THE END  MAKES            *00020650
CCC      THE CHOLESKY DECOMPOSITION.                                   *00020660
CCC                                                                    *00020670
CCC      ***************************************************************00020680
CCC                                                                     00020690
C                                                                       00020700
      IMPLICIT  REAL * 8 (A-H,O-Z)                                      00020710
C                                                                       00020720
      COMMON /ONE/ AJ(2,2),B(4,16),COPY(37,298),D(4,4),DL(4,4),         00020730
     1DNZR(2,8),ESM(16,16),F(24,298),SSM(298,66),STOR(20,150)           00020740
C                                                                       00020750
      COMMON /TWO/ AL(5),DENS(5),DETA(8),DXI(8),E(5),EPSO(4),           00020760
     1FE(16),GC(4),GX(4),POI(5),R(149),SHP(8),SHT(5),TK(5),Z(149)       00020770
C                                                                       00020780
      COMMON /TRE/ AXIALF,DTI,EVERY,FORS1,OMEGA,HTCI,HTCO,              00020790
     1POSI,PRES,Q1,Q2,Q3,SHMAX,SMAX,TIME,TIME1,TINIT,TMAXM,TREF,        00020800
     2TMAXO,RMAXM,RMAXO,ZMAXM,ZMAXO,Q4,BIG                              00020810
C                                                                       00020820
      COMMON /FOR/ NN(40,9)                                             00020830
C                                                                       00020840
      COMMON /FIV/ NCFI(37),NCFO(37),NNR(37),NRE(37),NNL(37),           00020850
     1NPN(37)                                                           00020860
      COMMON /SIX/ IBAND,IEXT,INTP,IVEC,JVEC,KK,LOAD,NCFIT,NCFOT,       00020870
     1NCHECK,NET,NDF,NGP,NNE,NNLT,NNRT,NPASS,NPROB,NRPPI,NRPPO,         00020880
     3NNT,NNRE,NPNT,IPLANE,NBAND                                        00020890
C                                                                       00020900
CC                                                                      00020910
C                                                                       00020920
      DO  5  I = 1 , NDF                                                00020930
      DO  5  J = 1 , IBAND                                              00020940
    5 SSM(I,J) = 0.DO                                                   00020950
      PAI = 3.141592653589793                                           00020960
      DO  31  L = 1 , NET                                               00020970
      IF  (L. EQ .1 )  GO  TO  1                                        00020980
      IF  (NN(L,9).EQ.NN(L-1,9))  GO  TO `3                             00020990
C                                                                       00021000
    1 K = NN(L,9)                                                       00021010
      EL = 1.DO - POI(K)                                                00021020
      FACT = (E(K)*EL)/((1.DO+POI(K))*(1.DO-2.DO*POI(K)))               00021030
```

```
      EL = POI(K) / EL                                                  00021040
C                                                                       00021050
      D(1,1) = FACT                                                     00021060
      D(1,2) = EL * FACT                                                00021070
      D(1,3) = D(1,2)                                                   00021080
      D(1,4) = 0.D0                                                     00021090
      D(2,2) = FACT                                                     00021100
      D(2,3) = D(1,2)                                                   00021110
      D(2,4) = 0.D0                                                     00021120
      D(3,3) = FACT                                                     00021130
      D(3,4) = 0.D0                                                     00021140
      D(4,4) = E(K)/(2.D0*(1.D0+POI(K)))                                00021150
C                                                                       00021160
C                                                                       00021170
      DO  27  I = 1 , 4                                                 00021180
      DO  27  J = I , 4                                                 00021190
   27 D(J,I) = D(I,J)                                                   00021200
C                                                                       00021210
C                                                                       00021220
      CALL  LLT  (4,D,DL)                                               00021230
C                                                                       00021240
C                                                                       00021250
C     DL  IS THE LOWER TRIANGLE OF DECOMPOSED D.                        00021260
C                                                                       00021270
C                                                                       00021280
    3 CONTINUE                                                          00021290
C                                                                       00021300
      NNE2 = 2 * NNE                                                    00021310
      DO  11  I = 1 , NNE2                                              00021320
      DO  11  J = 1 , NNE2                                              00021330
   11 ESM(I,J) = 0.D0                                                   00021340
C                                                                       00021350
C                                                                       00021360
      DO  9  N = 1,NGP                                                  00021370
      ETA = GX(N)                                                       00021380
      DO  8  M = 1,NGP                                                  00021390
      XI = GX(M)                                                        00021400
C                                                                       00021410
C                                                                       00021420
      DO  7  I = 1,4                                                    00021430
      DO  7  J = 1,NNE2                                                 00021440
    7 B(I,J) = 0.D0                                                     00021450
C                                                                       00021460
C                                                                       00021470
C                                                                       00021480
      SHP(1) = .25D0 * (1.D0-XI) * (1.D0-ETA) * (-XI-ETA-1.D0)          00021490
      SHP(2) = .5D0 * (1.D0-XI**2) * (1.D0-ETA)                         00021500
      SHP(3) = .25D0 * (1.D0+XI) * (1.D0-ETA) * (XI-ETA-1.D0)           00021510
```

```
      SHP(4) = .5D0 * (1.D0-ETA**2) * (1.D0+XI)                          00021520
      SHP(5) = .25D0 * (1.D0+XI) * (1.D0+ETA) * (XI+ETA-1.D0)            00021530
      SHP(6) = .5D0 * (1.D0-XI**2) * (1.D0+ETA)                          00021540
      SHP(7) = .25D0 * (1.D0-XI) * (1.D0+ETA) * (-XI+ETA-1.D0)           00021550
      SHP(8) = .5D0 * (1.D0-ETA**2) * (1.D0-XI)                          00021560
C                                                                        00021570
C                                                                        00021580
C     DERIVATIVES OF THE SHAPE FUNCTIONS WITH RESPECT TO XI.             00021590
C                                                                        00021600
      DXI(1)     = .25D0 * (1.D0-ETA) * (2.D0*XI+ETA)                    00021610
      DXI(2)     = -XI * (1.D0-ETA)                                      00021620
      DXI(3)     = .25D0 * (1.D0-ETA) * (2.D0*XI-ETA)                    00021630
      DXI(4)     = .5D0 * (1.D0-ETA**2)                                  00021640
      DXI(5)     = .25D0 * (1.D0+ETA) * (2.D0*XI+ETA)                    00021650
      DXI(6)     = -XI * (1.D0+ETA)                                      00021660
      DXI(7)     = .25D0 * (1.D0+ETA) * (2.D0*XI-ETA)                    00021670
      DXI(8)     = -.5D0 * (1.D0-ETA**2)                                 00021680
C                                                                        00021690
C                                                                        00021700
C     DERIVATIVES OF THE SHAPE FUNCTIONS WITH RESPECT TO ETA.            00021710
C                                                                        00021720
      DETA(1)    = .25D0 * (1.D0-XI) * (2.D0*ETA+XI)                     00021730
      DETA(2)    = -.5D0 * (1.D0-XI**2)                                  00021740
      DETA(3)    = .25D0 * (1.D0+XI) * (2.D0*ETA-XI)                     00021750
      DETA(4)    = -ETA * (1.D0+XI)                                      00021760
      DETA(5)    = .25D0 * (1.D0+XI) * (2.D0*ETA+XI)                     00021770
      DETA(6)    = .5D0 * (1.D0-XI**2)                                   00021780
      DETA(7)    = .25D0 * (1.D0-XI) * (2.D0*ETA-XI)                     00021790
      DETA(8)    = -ETA * (1.D0-XI)                                      00021800
C                                                                        00021810
C                                                                        00021820
C        CONSTRUCTION OF THE JACOBIAN MATRIX.                            00021830
C                                                                        00021840
      DO  20  I = 1 , 2                                                  00021850
      DO  20  J = 1 , 2                                                  00021860
   20 AJ(I,J) = 0.D0                                                     00021870
C                                                                        00021880
      DO  22  I = 1 , NNE                                                00021890
      AJ(1,1) = AJ(1,1) + DXI(I)  * Z(NN(L,I))                           00021900
      AJ(1,2) = AJ(1,2) + DXI(I)  * R(NN(L,I))                           00021910
      AJ(2,1) = AJ(2,1) + DETA(I) * Z(NN(L,I))                           00021920
      AJ(2,2) = AJ(2,2) + DETA(I) * R(NN(L,I))                           00021930
   22 CONTINUE                                                           00021940
C                                                                        00021950
```

```
C                                                                       00022000
      DET =AJ(1,1) * AJ(2,2) - AJ(1,2) * AJ(2,1)                        00022010
      DEI =1.0D0/DET                                                    00022020
C                                                                       00022030
CC                                                                      00022040
CC    INVERTING THE JACOBIAN AND STORING IN  AJ  AGAIN.                 00022050
CC                                                                      00022060
C                                                                       00022070
      DUM = AJ(1,1) * DEI                                               00022080
      AJ(1,1) = AJ(2,2) * DEI                                           00022090
      AJ(1,2) = -AJ(1,2) * DEI                                          00022100
      AJ(2,1) = -AJ(2,1) * DEI                                          00022110
      AJ(2,2) = DUM                                                     00022120
C                                                                       00022130
CC                                                                      00022140
C                                                                       00022150
      CR = 0.D0                                                         00022160
      DO  2  I = 1,NNE                                                  00022170
      M1 = NN(L,I)                                                      00022180
    2 CR = CR + SHP(I) * R(M1)                                          00022190
C                                                                       00022200
CC                                                                      00022210
CC    EVALUATION OF THE B MATRIX                                        00022220
CC                                                                      00022230
C                                                                       00022240
      DO  15  I = 1, NNE                                                00022250
      K1 = 2 * I - 1                                                    00022260
      K2 = K1 + 1                                                       00022270
      B(1,K2) = AJ(1,1) * DXI(I) + AJ(1,2) * DETA(I)                    00022280
      B(2,K1) = AJ(2,1) * DXI(I) + AJ(2,2) * DETA(I)                    00022290
      B(3,K1) = SHP(I) / CR                                             00022300
      B(4,K1) = B(1,K2)                                                 00022310
      B(4,K2) = B(2,K1)                                                 00022320
   15 CONTINUE                                                          00022330
C                                                                       00022340
CC                                                                      00022350
CC    CALCULATION OF (DL TRANSPOSE)*B AND STORING IN <B>                00022360
CC                                                                      00022370
C                                                                       00022380
      DO  16  I = 1 ,4                                                  00022390
      DO  13  J = 1,NNE2                                                00022400
      DUM = 0.0D0                                                       00022410
      DO  17  K = I,4                                                   00022420
   17 DUM = DUM + DL(K,I) * B(K,J)                                      00022430
   13 B(I,J) = DUM                                                      00022440
   16 CONTINUE                                                          00022450
C                                                                       00022460
C                                                                       00022470
```

```
C      NOTE : <B> NOW CONTAINS THE PRODUCT OF (DL TRANSPOSE)*B                00022480
CCCCC                                                                        00022490
CCCCC      CALCULATION OF <BT>*<D>*<B> AND ELEMENT STIFFNESS MATRIX           00022500
CCCCC                              <E S M >                                  00022510
CCC                                                                          00022520
                                                                             00022530
      DUM1 = 2.DO * PAI * CR * DET * GC(N) * GC(M)                           00022540
      DO  19   I = 1,NNE2                                                    00022550
      DO  19   J = 1, I                                                      00022560
      G = 0.0DO                                                              00022570
      DO  18   K = 1 ,4                                                      00022580
   18 G = G + B(K,I) * B(K,J)                                                00022590
      ESM(I,J) = ESM(I,J) + G * DUM1                                         00022600
   19 CONTINUE                                                               00022610
C                                                                            00022620
CC                                                                           00022630
      DO  24   I = 1 , NNE2                                                  00022640
      DO  24   J = 1 , I                                                     00022650
   24 ESM(J,I) = ESM(I,J)                                                    00022660
C                                                                            00022670
CC                                                                           00022680
    8 CONTINUE                                                               00022690
    9 CONTINUE                                                               00022700
CCCCC                                                                        00022710
CCCCC                                                                        00022720
CCCCC                                                                        00022730
CCCCC          CONSTRUCTION OF  THE SYSTEM STIFFNESS MATRIX                  00022740
CCCCC          IN BANDED FORM.                                               00022750
CCCCC                                                                        00022760
CCC                                                                          00022770
      DO  25   I = 1 , NNE                                                   00022780
      II = 2 * (NN(L,I) -1)                                                  00022790
C                                                                            00022800
      DO  25   J = 1 , NNE                                                   00022810
      JI = 2 * (NN(L,J) - 1)                                                 00022820
      IF  (II.GT.JI)  GO  TO  25                                             00022830
      JI = JI - II                                                           00022840
C                                                                            00022850
      DO  25   K = 1 , 2                                                     00022860
      JJ = II + K                                                            00022870
      L1 = 2 * (I-1) + K                                                     00022880
      IJ = 1                                                                 00022890
      IF  (JI.EQ.0)  IJ = K                                                  00022900
      DO  25   JI = IJ , 2                                                   00022910
      M1 = 2 * (J-1) + JI                                                    00022920
      KK = JI + JI - K + 1                                                   00022930
      SSM(JJ,KK) = SSM(JJ,KK) + ESM(L1,M1)                                   00022940
   25 CONTINUE                                                               00022950
```

```
C
   31 CONTINUE
C
      IF (IPLANE.EQ.0)  GO  TO  49
C
CCC
CCC      PART OF SSM IS BEING STORED IN COPY.
CCC
C
      DO  41  J = 1 , NDF
      DO  41  I = 1 , NNRE
   41 COPY(I,J) = 0.DO
C
C
      DO  44  II = 1 , NNRE
      K = 2 * NRE(II)
      JJ = K
      DO  42  J = 1 , IBAND
      IF  (JJ.GT.NDF)  GO  TO  45
      COPY(II,JJ) = SSM(K,J)
   42 JJ = JJ + 1
C
C
   45 LL = K - 1
      IF  (LL.EQ.0)  GO  TO  44
      JJ = LL
C
C
      DO  43  J = 1 , LL
      IF  (J.EQ.IBAND)  GO  TO  44
      COPY(II,JJ) = SSM(JJ,J+1)
   43 JJ = JJ - 1
   44 CONTINUE
C
CCC
C
   49 CONTINUE
C
      IF  (NNRT.EQ.0)  GO  TO  33
      DO  34  I = 1 , NNRT
      K = 2 * NNR(I) - 1
   34 SSM(K,1) = SSM(K,1) * BIG
C
   33 IF  (NNLT.EQ.0)  GO  TO  35
      DO  36  I = 1 , NNLT
      K = 2 * NNL(I)
   36 SSM(K,1) = SSM(K,1) * BIG
   35 CONTINUE
```

00022960
00022970
00022980
00022990
00023000
00023010
00023020
00023030
00023040
00023050
00023060
00023070
00023080
00023090
00023100
00023110
00023120
00023130
00023140
00023150
00023160
00023170
00023180
00023190
00023200
00023210
00023220
00023230
00023240
00023250
00023260
00023270
00023280
00023290
00023300
00023310
00023320
00023330
00023340
00023350
00023360
00023370
00023380
00023390
00023400
00023410
00023420
00023430

```
C                                                                       00023440
CCCCC                                                                   00023450
CCCCC                                                                   00023460
C          DECOMPOSING SYSTEM STIFFNESS MATRIX                          00023470
C                                                                       00023480
      DO  300  I = 1 , NDF                                              00023490
      DUM = SSM(I,1)                                                    00023500
      DUM = DSQRT(DUM)                                                  00023510
      DO  200  J = 1 , IBAND                                            00023520
  200 SSM(I,J) = SSM(I,J) / DUM                                         00023530
C                                                                       00023540
      DO  260  J = 2 , IBAND                                            00023550
      L = I + J -1                                                      00023560
      IF  (L.GT.NDF)  GO  TO  260                                       00023570
      AA = SSM(I,J)                                                     00023580
      IF  (AA.EQ.0.DO)  GO  TO  260                                     00023590
      DO  250  K = J , IBAND                                            00023600
      M = K - J + 1                                                     00023610
  250 SSM(L,M) = SSM(L,M) - AA * SSM(I,K)                               00023620
  260 CONTINUE                                                          00023630
  300 CONTINUE                                                          00023640
  350 RETURN                                                            00023650
      END                                                               00023660
```

```
      SUBROUTINE  FORMF                                                     00023670
C                                                                           00023680
C                                                                           00023690
CCCCCCCCCCC   ************************************************************  00023700
CCCCCCCCCCC                                                               *  00023710
CCCCCCCCCCC   THE THERMAL LOAD VECTORS ARE BEING CALCULATED HERE          *  00023720
CCCCCCCCCCC   AND FOR THE BOUNDARY CONDITION "ENDS REMAIN PLANE"          *  00023730
CCCCCCCCCCC   ANOTHER LOAD VECTOR FOR THE UNIT END  DISPLACEMENT          *  00023740
CCCCCCCCCCC   IS BEING CALCULATED.                                        *  00023750
CCCCCCCCCCC                                                               *  00023760
CCCCCCCCCCC   ************************************************************  00023770
C                                                                           00023780
C                                                                           00023790
      IMPLICIT  REAL * 8 (A-H,O-Z)                                          00023800
C                                                                           00023810
      COMMON /ONE/ AJ(2,2),B(4,16),COPY(37,298),D(4,4),DL(4,4),             00023820
     1DNZR(2,8),ESM(16,16),F(24,298),SSM(298,66),STOR(20,150)               00023830
C                                                                           00023840
      COMMON /TWO/ AL(5),DENS(5),DETA(8),DXI(8),E(5),EPSO(4),               00023850
     1FE(16),GC(4),GX(4),POI(5),R(149),SHP(8),SHT(5),TK(5),Z(149)           00023860
C                                                                           00023870
      COMMON /TRE/ AXIALF,DTI,EVERY,FORS1,OMEGA,HTCI,HTCO,                  00023880
     1POSI,PRES,Q1,Q2,Q3,SHMAX,SMAX,TIME,TIME1,TINIT,TMAXM,TREF,            00023890
     2TMAXO,RMAXM,RMAXO,ZMAXM,ZMAXO,Q4,BIG                                  00023900
C                                                                           00023910
      COMMON /FOR/ NN(40,9)                                                 00023920
C                                                                           00023930
      COMMON /FIV/ NCFI(37),NCFO(37),NNR(37),NRE(37),NNL(37),               00023940
     1NPN(37)                                                               00023950
      COMMON /SIX/ IBAND,IEXT,INTP,IVEC,JVEC,KK,LOAD,NCFIT,NCFOT,           00023960
     1NCHECK,NET,NDF,NGP,NNE,NNLT,NNRT,NPASS,NPROB,NRPPI,NRPPO,             00023970
     3NNT,NNRE,NPNT,IPLANE,NBAND                                            00023980
C                                                                           00023990
      PAI = 3.1415926535897 93                                              00024000
      IDUM = IVEC + JVEC                                                    00024010
      IF  (LOAD.NE.0)  IDUM = IDUM -1                                       00024020
      DO  1  I = 1 , IDUM                                                   00024030
      DO  1  J = 1 , NDF                                                    00024040
    1 F(I,J) = 0.DO                                                         00024050
C                                                                           00024060
      IDUM = IDUM + 1                                                       00024070
      IF  (LOAD.NE.0)  IDUM = IDUM +1                                       00024080
      DO  4  I = 1 , NDF                                                    00024090
    4 F(IDUM,I) = 0.DO                                                      00024100
C                                                                           00024110
      NNE2 = 2 * NNE                                                        00024120
```

```
C
      DO  25  L = 1 , NET
CCC
CCC
      DO  9  N = 1,NGP
      ETA = GX(N)
      DO  8  M = 1,NGP
      XI = GX(M)
C
CC
      DO  7  I = 1,4
      DO  7  J = 1,NNE2
    7 B(I,J) = 0.D0
C
CCC
      SHP(1) = .25D0 * (1.D0-XI) * (1.D0-ETA) * (-XI-ETA-1.D0)
      SHP(2) = .5D0 * (1.D0-XI**2) * (1.D0-ETA)
      SHP(3) = .25D0 * (1.D0+XI) * (1.D0-ETA) * (XI-ETA-1.D0)
      SHP(4) = .5D0 * (1.D0-ETA**2) * (1.D0+XI)
      SHP(5) = .25D0 * (1.D0+XI) * (1.D0+ETA) * (XI+ETA-1.D0)
      SHP(6) = .5D0 * (1.D0-XI**2) * (1.D0+ETA)
      SHP(7) = .25D0 * (1.D0-XI) * (1.D0+ETA) * (-XI+ETA-1.D0)
      SHP(8) = .5D0 * (1.D0-ETA**2) * (1.D0-XI)
CCCCC
       DERIVATIVES OF THE SHAPE FUNCTIONS WITH RESPECT TO XI.
CCCCC
      DXI(1)     = .25D0 * (1.D0-ETA) * (2.D0*XI+ETA)
      DXI(2)     = -XI * (1.D0-ETA)
      DXI(3)     = .25D0 * (1.D0-ETA) * (2.D0*XI-ETA)
      DXI(4)     = .5D0 * (1.D0-ETA**2)
      DXI(5)     = .25D0 * (1.D0+ETA) * (2.D0*XI+ETA)
      DXI(6)     = -XI * (1.D0+ETA)
      DXI(7)     = .25D0 * (1.D0+ETA) * (2.D0*XI-ETA)
      DXI(8)     = -.5D0 * (1.D0-ETA**2)
C
CCCCC
       DERIVATIVES OF THE SHAPE FUNCTIONS WITH RESPECT TO ETA.
CCCCC
      DETA(1)    = .25D0 * (1.D0-XI) * (2.D0*ETA+XI)
      DETA(2)    = -.5D0 * (1.D0-XI**2)
      DETA(3)    = .25D0 * (1.D0+XI) * (2.D0*ETA-XI)
      DETA(4)    = -ETA * (1.D0+XI)
      DETA(5)    = .25D0 * (1.D0+XI) * (2.D0*ETA+XI)
```

00024130
00024140
00024150
00024160
00024170
00024180
00024190
00024200
00024210
00024220
00024230
00024240
00024250
00024260
00024270
00024280
00024290
00024300
00024310
00024320
00024330
00024340
00024350
00024360
00024370
00024380
00024390
00024400
00024410
00024420
00024430
00024440
00024450
00024460
00024470
00024480
00024490
00024500
00024510
00024520
00024530
00024540
00024550
00024560
00024570
00024580
00024590
00024600

```
      DETA(6)     = .5D0 * (1.D0-XI**2)                                    00024610
      DETA(7)     = .25D0 * (1.D0-XI) * (2.D0*ETA-XI)                      00024620
      DETA(8)     = -ETA * (1.D0-XI)                                       00024630
C                                                                          00024640
CCC                                                                        00024650
CCC      CONSTRUCTION OF THE JACOBIAN MATRIX.                              00024660
CCC                                                                        00024670
C                                                                          00024680
      DO  20  I = 1 , 2                                                    00024690
      DO  20  J = 1 , 2                                                    00024700
   20 AJ(I,J) = 0.D0                                                       00024710
C                                                                          00024720
CC                                                                         00024730
      DO  22  I = 1 , NNE                                                  00024740
      AJ(1,1) = AJ(1,1) + DXI(I)  * Z(NN(L,I))                             00024750
      AJ(1,2) = AJ(1,2) + DXI(I)  * R(NN(L,I))                             00024760
      AJ(2,1) = AJ(2,1) + DETA(I) * Z(NN(L,I))                             00024770
      AJ(2,2) = AJ(2,2) + DETA(I) * R(NN(L,I))                             00024780
   22 CONTINUE                                                             00024790
C                                                                          00024800
      DET =AJ(1,1) * AJ(2,2) - AJ(1,2) * AJ(2,1)                           00024810
      DEI =1.0D0/DET                                                       00024820
C                                                                          00024830
CCC                                                                        00024840
CCC   INVERTING THE JACOBIAN AND STORING IN  AJ  AGAIN.                    00024850
CCC                                                                        00024860
C                                                                          00024870
      DUM = AJ(1,1) * DEI                                                  00024880
      AJ(1,1) = AJ(2,2) * DEI                                              00024890
      AJ(1,2) = -AJ(1,2) * DEI                                             00024900
      AJ(2,1) = -AJ(2,1) * DEI                                             00024910
      AJ(2,2) = DUM                                                        00024920
C                                                                          00024930
CC                                                                         00024940
      CR = 0.D0                                                            00024950
      DO  2  I = 1,NNE                                                     00024960
      M1 = NN(L,I)                                                         00024970
    2 CR = CR + SHP(I) * R(M1)                                             00024980
C                                                                          00024990
CCC                                                                        00025000
CCC                                                                        00025010
CCC   EVALUATION OF THE B MATRIX                                           00025020
CCC                                                                        00025030
C                                                                          00025040
      DO  15  I = 1, NNE                                                   00025050
      K1 = 2 * I - 1                                                       00025060
      K2 = K1 + 1                                                          00025070
      B(1,K2) = AJ(1,1) * DXI(I) + AJ(1,2) * DETA(I)                       00025080
```

```
      B(2,K1) = AJ(2,1) * DXI(I) + AJ(2,2) * DETA(I)                    00025090
      B(3,K1) = SHP(I) / CR                                             00025100
      B(4,K1) = B(1,K2)                                                 00025110
      B(4,K2) = B(2,K1)                                                 00025120
   15 CONTINUE                                                          00025130
C                                                                       00025140
C                                                                       00025150
      DUM1 = 2.D0 * PAI * CR * DET * GC(N) * GC(M)                      00025160
C                                                                       00025170
C                                                                       00025180
      DO  30  KK = 1 , IVEC                                             00025190
C                                                                       00025200
CCCC                                                                    00025210
CCC      EVALUATING TEMPERATURE AT GAUSS POINT                          00025220
C                                                                       00025230
      TGP = 0.D0                                                        00025240
      DO  3  I = 1 , NNE                                                00025250
      M1 = NN(L , I)                                                    00025260
    3 TGP = TGP + SHP(I) * (STOR(KK,M1)-TREF)                           00025270
C                                                                       00025280
      TGP = AL(NN(L,9)) * TGP                                           00025290
C                                                                       00025300
CCCCC                                                                   00025310
CCCCC    CALCULATION OF ELEMENT INITIAL STRAIN VECTOR <EPSO>            00025320
CCCCC                                                                   00025330
C                                                                       00025340
      DO  6  I = 1 , 3                                                  00025350
    6 EPSO(I) = TGP                                                     00025360
C                                                                       00025370
CCCCCC                                                                  00025380
CCCCCC   NOTE  EPSO(4) =0.D0                                            00025390
CCCCCC                                                                  00025400
CCCCCC   PRODUCT OF <BTRANSPOSE> * <D> * <EPSO>                         00025410
CCCCCC   CALCULATION OF ELEMENT LOAD VECTOR ,                           00025420
C                                                                       00025430
      DUM = (TGP*E(NN(L,9)))/(1.D0-2.D0*POI(NN(L,9)))                   00025440
C                                                                       00025450
      DUM = DUM * DUM1                                                  00025460
      DO  17  I = 1 , NNE2 , 2                                          00025470
      J = I + 1                                                         00025480
      FE(I) = (B(2,I) + B(3,I)) * DUM                                   00025490
   17 FE(J) = B(1,J) * DUM                                              00025500
C                                                                       00025510
CCCCC                                                                   00025520
CCCCC    CONSTRUCTION OF SYSTEM LOAD VECTORS   F                        00025530
CCCCC                                                                   00025540
C                                                                       00025550
      DO  18  I = 1 , NNE                                               00025560
```

```
      I2 = 2 * I - 1                                                    00025570
      I1 = NN(L,I) * 2 - 1                                              00025580
      F(KK,I1) = F(KK,I1) + FE(I2)                                      00025590
      F(KK,I1+1) = F(KK,I1+1) + FE(I2+1)                                00025600
C                                                                       00025610
C                                                                       00025620
   18 CONTINUE                                                          00025630
   30 CONTINUE                                                          00025640
    8 CONTINUE                                                          00025650
    9 CONTINUE                                                          00025660
   25 CONTINUE                                                          00025670
C                                                                       00025680
C                                                                       00025690
      IF (IPLANE.EQ.0)  GO  TO  40                                      00025700
C                                                                       00025710
C        LOAD VECTOR FOR UNIT END DISPLACEMENT                          00025720
C                                                                       00025730
      K = IDUM                                                          00025740
C                                                                       00025750
      DO  41  I = 1 , NNRE                                              00025760
      L = 2 * NRE(I)                                                    00025770
   41 F(K,L) = COPY(I,L) * BIG                                          00025780
C                                                                       00025790
   40 CONTINUE                                                          00025800
      RETURN                                                            00025810
      END                                                               00025820
C                                                                       00025830
C                                                                       00025840
```

```
      SUBROUTINE  CENTF                                                     00025850
                                                                            00025860
C                                                                           00025870
CCCCCCCCC*************************************************************    00025880
CCCCCCCCC                                                             *     00025890
CCCCCCCCC  SUBROUTINE CENTF  CALCULATES THE CENTRIFUGAL LOAD          *     00025900
CCCCCCCCC  VECTOR FOR A GIVEN NUMBER OF REVOLUTIONS PER MINUTE        *     00025910
CCCCCCCCC                                                             *     00025920
CCCCCCCCC*************************************************************    00025930
                                                                            00025940
C                                                                           00025950
      IMPLICIT  REAL * 8 (A-H,O-Z)                                          00025960
                                                                            00025970
CC                                                                          00025980
      COMMON /ONE/ AJ(2,2),B(4,16),COPY(37,298),D(4,4),DL(4,4),             00025990
     1DNZR(2,8),ESM(16,16),F(24,298),SSM(298,66),STOR(20,150)               00026000
                                                                            00026010
C                                                                           00026020
      COMMON /TWO/ AL(5),DENS(5),DETA(8),DXI(8),E(5),EPSO(4),               00026030
     1FE(16),GC(4),GX(4),PCI(5),R(149),SHP(8),SHT(5),TK(5),Z(149)           00026040
C                                                                           00026050
      COMMON /TRE/ AXIALF,DTI,EVERY,FORS1,OMEGA,HTCI,HTCO,                  00026060
     1POSI,PRES,Q1,Q2,Q3,SHMAX,SMAX,TIME,TIME1,TINIT,TMAXM,TREF,            00026070
     2TMAXO,RMAXM,RMAXO,ZMAXM,ZMAXO,Q4,BIG                                  00026080
C                                                                           00026090
      COMMON /FOR/ NN(40,9)                                                 00026100
C                                                                           00026110
      COMMON /FIV/ NCFI(37),NCFO(37),NNR(37),NRE(37),NNL(37),               00026120
     1NPN(37)                                                               00026130
      COMMON /SIX/ IBAND,IEXT,INTP,IVEC,JVEC,KK,LOAD,NCFIT,NCFOT,           00026140
     1NCHECK,NET,NDF,NGP,NNE,NNET,NNRT,NPASS,NPROB,NRPPI,NRPPO,             00026150
     3NNT,NNRE,NPNT,IPLANE,NBAND                                            00026160
C                                                                           00026170
      DATA  BRIT/'BRIT'/                                                    00026180
C                                                                           00026190
      PAI = 3.1415926535897933                                              00026200
      II = IVEC + 1                                                         00026210
C                                                                           00026220
CC                                                                          00026230
      DO  25  L = 1 , NET                                                   00026240
      NNE2 = 2 * NNE                                                        00026250
      DO  11  I = 1 , NNE2 , 2                                              00026260
   11 FE(I) = 0.D0                                                          00026270
C                                                                           00026280
CCC                                                                         00026290
C                                                                           00026300
      K = NN(L,9)
```

```
      DUM1 = 2.D0 * PAI * DENS(K) * OMEGA**2                                    00026310
C                                                                               00026320
      IF  (Q1.EQ.BRIT)  DUM1 = DUM1 / 386.0885827                               00026330
      IF  (Q1.NE.BRIT)  DUM1 = DUM1 / 980.665                                   00026340
C                                                                               00026350
      DO  9  N = 1,NGP                                                          00026360
      ETA = GX(N)                                                               00026370
      DO  8  M = 1,NGP                                                          00026380
      XI = GX(M)                                                                00026390
C                                                                               00026400
C                                                                               00026410
      SHP(1) = .25D0 * (1.D0-XI) * (1.D0-ETA) * (-XI-ETA-1.D0)                  00026420
      SHP(2) = .5D0 * (1.D0-XI**2) * (1.D0-ETA)                                 00026430
      SHP(3) = .25D0 * (1.D0+XI) * (1.D0-ETA) * (XI-ETA-1.D0)                   00026440
      SHP(4) = .5D0 * (1.D0-ETA**2) * (1.D0+XI)                                 00026450
      SHP(5) = .25D0 * (1.D0+XI) * (1.D0+ETA) * (XI+ETA-1.D0)                   00026460
      SHP(6) = .5D0 * (1.D0-XI**2) * (1.D0+ETA)                                 00026470
      SHP(7) = .25D0 * (1.D0-XI) * (1.D0+ETA) * (-XI+ETA-1.D0)                  00026480
      SHP(8) = .5D0 * (1.D0-ETA**2) * (1.D0-XI)                                 00026490
C                                                                               00026500
CC                                                                              00026510
C     DERIVATIVES OF THE SHAPE FUNCTIONS WITH RESPECT TO XI.                    00026520
                                                                                00026530
C                                                                               00026540
      DXI(1)     = .25D0 * (1.D0-ETA) * (2.D0*XI+ETA)                           00026550
      DXI(2)     = -XI * (1.D0-ETA)                                             00026560
      DXI(3)     = .25D0 * (1.D0-ETA) * (2.D0*XI-ETA)                           00026570
      DXI(4)     = .5D0 * (1.D0-ETA**2)                                         00026580
      DXI(5)     = .25D0 * (1.D0+ETA) * (2.D0*XI+ETA)                           00026590
      DXI(6)     = -XI * (1.D0+ETA)                                             00026600
      DXI(7)     = .25D0 * (1.D0+ETA) * (2.D0*XI-ETA)                           00026610
      DXI(8)     = -.5D0 * (1.D0-ETA**2)                                        00026620
C                                                                               00026630
CCCC                                                                            00026640
C     DERIVATIVES OF THE SHAPE FUNCTIONS WITH RESPECT TO ETA.                   00026650
CCC                                                                             00026660
                                                                                00026670
      DETA(1)    = .25D0 * (1.D0-XI) * (2.D0*ETA+XI)                            00026680
      DETA(2)    = -.5D0 * (1.D0-XI**2)                                         00026690
      DETA(3)    = .25D0 * (1.D0+XI) * (2.D0*ETA-XI)                            00026700
      DETA(4)    = -ETA * (1.D0+XI)                                             00026710
      DETA(5)    = .25D0 * (1.D0+XI) * (2.D0*ETA+XI)                            00026720
      DETA(6)    = .5D0 * (1.D0-XI**2)                                          00026730
      DETA(7)    = .25D0 * (1.D0-XI) * (2.D0*ETA-XI)                            00026740
      DETA(8)    = -ETA * (1.D0-XI)                                             00026750
C                                                                               00026760
CCC                                                                             00026770
C        CONSTRUCTION OF THE JACOBIAN MATRIX.                                   00026780
```

```
C                                                                       00026790
C                                                                       00026800
      DO  20  I = 1 , 2                                                 00026810
      DO  20  J = 1 , 2                                                 00026820
   20 AJ(I,J) = 0.D0                                                    00026830
C                                                                       00026840
C                                                                       00026850
      DO  22  I = 1 , NNE                                               00026860
      AJ(1,1) = AJ(1,1) + DXI(I)  * Z(NN(L,I))                         00026870
      AJ(1,2) = AJ(1,2) + DXI(I)  * R(NN(L,I))                         00026880
      AJ(2,1) = AJ(2,1) + DETA(I) * Z(NN(L,I))                         00026890
      AJ(2,2) = AJ(2,2) + DETA(I) * R(NN(L,I))                         00026900
   22 CONTINUE                                                          00026910
C                                                                       00026920
C                                                                       00026930
      DET =AJ(1,1) * AJ(2,2) - AJ(1,2) * AJ(2,1)                        00026940
      DEI =1.0D0/DET                                                    00026950
C                                                                       00026960
C                                                                       00026970
C     INVERTING THE JACOBIAN AND STORING IN  AJ  AGAIN.                 00026980
C                                                                       00026990
C                                                                       00027000
      DUM = AJ(1,1) * DEI                                               00027010
      AJ(1,1) = AJ(2,2) * DEI                                           00027020
      AJ(1,2) = -AJ(1,2) * DEI                                          00027030
      AJ(2,1) = -AJ(2,1) * DEI                                          00027040
      AJ(2,2) = DUM                                                     00027050
C                                                                       00027060
C                                                                       00027070
      CR = 0.D0                                                         00027080
      DO  2  I = 1,NNE                                                  00027090
      M1 = NN(L,I)                                                      00027100
    2 CR = CR + SHP(I) * R(M1)                                          00027110
      CR = CR**2                                                        00027120
C                                                                       00027130
C                                                                       00027140
      DUM = DUM1 * CR * DET * GC(N) * GC(M)                             00027150
C                                                                       00027160
C                                                                       00027170
      DO  17  I = 1 , NNE                                               00027180
      K = 2 * I - 1                                                     00027190
      FE(K) = FE(K) + SHP(I) * DUM                                      00027200
   17 CONTINUE                                                          00027210
C                                                                       00027220
C                                                                       00027230
    8 CONTINUE                                                          00027240
    9 CONTINUE                                                          00027250
C                                                                       00027260
```

```
C                                                                       00027270
CC          CONSTRUCTION OF SYSTEM CENTRIFUGAL LOAD VECTOR              00027280
CC                                                                      00027290
C                                                                       00027300
      DO  18  I = 1 , NNE                                               00027310
      I2 = 2 * I - 1                                                    00027320
      I1 = NN(L,I) * 2 - 1                                              00027330
      F(I1,I1) = F(I1,I1) + FE(I2)                                      00027340
   18 CONTINUE                                                          00027350
C                                                                       00027360
CC                                                                      00027370
   25 CONTINUE                                                          00027380
C                                                                       00027390
CC                                                                      00027400
      RETURN                                                            00027410
      END                                                               00027420
C                                                                       00027430
```

```
      SUBROUTINE  PRESS                                                     00027440
C                                                                           00027450
CCC                                                                         00027460
CCC   ***************************************************************     00027470
CCC                                                                   *     00027480
CCC   SUBROUTINE PRESS EVALUATES THE PRESSURE LOAD VECTOR             *     00027490
CCC   FOR THE SYSTEM WHEN THERE IS ANY PRESSURE (NE.0.D0)             *     00027500
CCC   ACTING ON THE BODY                                              *     00027510
CCC                                                                   *     00027520
CCC   ***************************************************************     00027530
C                                                                           00027540
      IMPLICIT  REAL * 8 (A-H,O-Z)                                          00027550
C                                                                           00027560
C                                                                           00027570
      COMMON /ONE/ AJ(2,2),B(4,16),COPY(37,298),D(4,4),DL(4,4),             00027580
     1DNZR(2,8),ESM(16,16),F(24,298),SSM(298,66),STOR(20,150)               00027590
C                                                                           00027600
      COMMON /TWO/ AL(5),DENS(5),DETA(8),DXI(8),E(5),EPSO(4),               00027610
     1FE(16),GC(4),GX(4),POI(5),R(149),SHP(8),SHT(5),TK(5),Z(149)           00027620
C                                                                           00027630
      COMMON /TRE/ AXIALF,DTI,EVERY,FORSI,OMEGA,HTCI,HTCO,                  00027640
     1POSI,PRES,Q1,Q2,Q3,SHMAX,SMAX,TIME,TIME1,TINIT,TMAXM,TREF,            00027650
     2TMAXO,RMAXM,RMAXO,ZMAXM,ZMAXO,Q4,BIG                                  00027660
C                                                                           00027670
      COMMON /FOR/ NN(40,9)                                                 00027680
C                                                                           00027690
      COMMON /FIV/ NCFI(37),NCFO(37),NNR(37),NRE(37),NNL(37),               00027700
     1NPN(37)                                                               00027710
      COMMON /SIX/ IBAND,IEXT,INTP,IVEC,JVEC,KK,LOAD,NCFIT,NCFOT,           00027720
     1NCHECK,NET,NDF,NGP,NNE,NNLT,NNRT,NPASS,NPROB,NRPPI,NRPPO,             00027730
     3NNT,NNRE,NPNT,IPLANE,NBAND                                            00027740
C                                                                           00027750
C                                                                           00027760
      PAI = 3.141592653589793                                               00027770
      L = 1                                                                 00027780
      IF  (OMEGA.NE.0.D0)  L = L + 1                                        00027790
      II = IVEC + L                                                         00027800
C                                                                           00027810
CCC                                                                         00027820
C                                                                           00027830
      NEUP = (NPNT-1) / 2                                                   00027840
C                                                                           00027850
CC       NEUP  IS THE NUMBER OF ELEMENTS UNDER PRESSURE                     00027860
C                                                                           00027870
                                                                            00027880
                                                                            00027890
```

```
      FACT = 2.D0 * PAI * PRES                                          00027900
C                                                                       00027910
C                                                                       00027920
      DO  10  L = 1 , NEUP                                              00027930
      K = 2 * L - 1                                                     00027940
      DO  2  I = 1 , NNE                                                00027950
    2 FE(I) = 0.D0                                                      00027960
C                                                                       00027970
C                                                                       00027980
      DO  9  N = 1 , NGP                                                00027990
      XI = GX(N)                                                        00028000
C                                                                       00028010
C                                                                       00028020
      SHP(1) = -.5D0 * (1.D0 - XI) * XI                                 00028030
      SHP(2) = 1.D0 - XI**2                                             00028040
      SHP(3) =  .5D0 * (1.D0 + XI) * XI                                 00028050
C                                                                       00028060
C                                                                       00028070
      DXI(1) = XI - .5D0                                                00028080
      DXI(2) = -2.D0 * XI                                               00028090
      DXI(3) = XI + .5D0                                                00028100
C                                                                       00028110
C                                                                       00028120
      RR = 0.D0                                                         00028130
      DNZ = 0.D0                                                        00028140
      DNR = 0.D0                                                        00028150
C                                                                       00028160
C                                                                       00028170
      DO  3  I = 1 , 3                                                  00028180
      J = NPN(K+I-1)                                                    00028190
      RR  = RR  + SHP(I) * R(J)                                         00028200
      DNZ = DNZ + DXI(I) * Z(J)                                         00028210
      DNR = DNR + DXI(I) * R(J)                                         00028220
    3 CONTINUE                                                          00028230
C                                                                       00028240
C                                                                       00028250
      DO  4  I = 1 , 3                                                  00028260
      J1 = 2 * I - 1                                                    00028270
      J2 = 2 * I                                                        00028280
      FE(J1) = FE(J1) + RR * DNZ * SHP(I) * GC(N)                       00028290
      FE(J2) = FE(J2) - RR * DNR * SHP(I) * GC(N)                       00028300
    4 CONTINUE                                                          00028310
    9 CONTINUE                                                          00028320
C                                                                       00028330
CCCC                                                                    00028340
CCCC      CONSTRUCTION OF SYSTEM PRESSURE LOAD VECTOR                   00028350
CCCC                                                                    00028360
C                                                                       00028370
```

```
      DO 5 I=1,3                                                        00028380
      J1=2*I-1                                                          00028390
      J2=2*I                                                            00028400
      J =2 * NPN(K+I-1) - 1                                             00028410
      F(II,J) = F(II,J) + FE (J1)* FACT                                 00028420
      F(II,J+1) = F(II,J+1) + FE(J2) * FACT                             00028430
    5 CONTINUE                                                          00028440
   10 CONTINUE                                                          00028450
C                                                                       00028460
C                                                                       00028470
      RETURN                                                            00028480
      END                                                               00028490
```

```
      SUBROUTINE  DISPL                                                     00028500
C                                                                           00028510
CCCCC                                                                       00028520
CCCCC    ************************************************************     00028530
CCCCC                                                                *     00028540
CCCCC    THIS SUBROUTINE SOLVES FOR DISPLACEMENTS,ONCE GIVEN         *     00028550
CCCCC    THE DECOMPOSED SYSTEM STIFFNESS MATRIX AND THE LOAD         *     00028560
CCCCC    VECTORS.      THE LOAD VECTORS ARE REPLACED BY THE          *     00028570
CCCCC    SOLUTIONS  'THE DISPLACEMENTS' .                            *     00028580
CCCCC                                                                *     00028590
CCCCC    ************************************************************     00028600
CCC                                                                         00028610
C                                                                           00028620
      IMPLICIT  REAL * 8 (A-H,O-Z)                                          00028630
C                                                                           00028640
      COMMON /ONE/ AJ(2,2),B(4,16),COPY(37,298),D(4,4),DL(4,4),             00028650
     1DNZR(2,8),ESM(16,16),F(24,298),SSM(298,66),STOR(20,150)               00028660
C                                                                           00028670
      COMMON /TWO/ AL(5),DENS(5),DETA(8),DXI(8),E(5),EPSO(4),               00028680
     1FE(16),GC(4),GX(4),POI(5),R(149),SHP(8),SHT(5),TK(5),Z(149)           00028690
C                                                                           00028700
      COMMON /TRE/ AXIALF,DTI,EVERY,FORS1,OMEGA,HTCI,HTCO,                  00028710
     1POSI,PRES,Q1,Q2,Q3,SHMAX,SMAX,TIME,TIME1,TINIT,TMAXM,TREF,            00028720
     2TMAXO,RMAXM,RMAXO,ZMAXM,ZMAXO,Q4,BIG                                  00028730
C                                                                           00028740
      COMMON /FOR/ NN(40,9)                                                 00028750
C                                                                           00028760
      COMMON /FIV/ NCFI(37),NCFO(37),NNR(37),NRE(37),NNL(37),               00028770
     1NPN(37)                                                               00028780
      COMMON /SIX/ IBAND,IEXT,INTP,IVEC,JVEC,KK,LOAD,NCFIT,NCFOT,           00028790
     1NCHECK,NET,NDF,NGP,NNE,NNLT,NNRT,NPASS,NPROB,NRPPI,NRPPO,             00028800
     3NNT,NNRE,NPNT,IPLANE,NBAND                                            00028810
CC                                                                          00028820
      ZERO = 0.DO                                                           00028830
      II = IVEC + JVEC + 1                                                  00028840
      IF (IPLANE.EQ.0)  II = II - 1                                         00028850
      DO  500  LL = 1 , II                                                  00028860
CCC                                                                         00028870
CCC                                                                         00028880
      DO  200  I = 1 , NDF                                                  00028890
      F(LL,I) = F(LL,I) / SSM(I,1)                                          00028900
      DO  100  J = 2 , IBAND                                                00028910
      L = I + J - 1                                                         00028920
      IF  (L. GT .NDF)  GO  TO  200                                         00028930
                                                                            00028940
                                                                            00028950
```

```
      DUM = SSM(I,J)                                                    00028960
  100 F(LL,L) = F(LL,L) - DUM * F(LL,I)                                 00028970
  200 CONTINUE                                                          00028980
C                                                                       00028990
CC                                                                      00029000
C                                                                       00029010
      F(LL,NDF) = F(LL,NDF) / SSM(NDF,1)                                00029020
      DO  400  L = 2 , NDF                                              00029030
      K = NDF - L + 1                                                   00029040
      SUM = ZERO                                                        00029050
      DO  300  J = 2 , IBAND                                            00029060
      M = J + K - 1                                                     00029070
      IF  (NDF. LT .M)  GO  TO  400                                     00029080
  300 SUM = SUM + SSM(K,J) * F(LL,M)                                    00029090
  400 F(LL,K) = (F(LL,K) - SUM) / SSM(K,1)                              00029100
C                                                                       00029110
CC                                                                      00029120
C                                                                       00029130
  500 CONTINUE                                                          00029140
C                                                                       00029150
CC                                                                      00029160
      IF (IPLANE.EQ.0)  GO  TO  20                                      00029170
C                                                                       00029180
CCCCC                                                                   00029190
         CORRECTING THE DISPLACEMENTS                                   00029200
CCCCC                                                                   00029210
C                                                                       00029220
      FORS1 = 0.D0                                                      00029230
      DO  7  I = 1 , NNRE                                               00029240
      DUM = 0.D0                                                        00029250
      DO  6  J = 1 , NDF                                                00029260
    6 DUM = DUM + COPY(I,J) * F(II,J)                                   00029270
    7 FORS1 = FORS1 + DUM                                               00029280
C                                                                       00029290
CC                                                                      00029300
      IF  (JVEC.EQ.0)  GO  TO  10                                       00029310
C                                                                       00029320
      DO  8  I = 1 , JVEC                                               00029330
      SUM = 0.D0                                                        00029340
      DO  9  J = 1 , NNRE                                               00029350
      K = 2 * NRE(J)                                                    00029360
      SUM = SUM + COPY(J,K) * F(IVEC+I,K) * BIG                         00029370
    9 CONTINUE                                                          00029380
C                                                                       00029390
      AXIALF = AXIALF + SUM                                             00029400
C                                                                       00029410
    8 CONTINUE                                                          00029420
   10 CONTINUE                                                          00029430
```

```
C                                                                       00029440
C                                                                       00029450
      DO  11  I = 1 , IVEC                                              00029460
      SUM = 0.DO                                                        00029470
      DO  12  J = 1 , NNRE                                              00029480
      K = 2 * NRE(J)                                                    00029490
      SUM = SUM + COPY(J,K) * F(I,K) * BIG                              00029500
   12 CONTINUE                                                          00029510
C                                                                       00029520
      FACT = (AXIALF+ SUM)/FORS1                                        00029530
C                                                                       00029540
      DO  28  L = 1 , NDF                                               00029550
      F(I,L) = F(I,L) + FACT * F(II,L)                                  00029560
   28 CONTINUE                                                          00029570
   11 CONTINUE                                                          00029580
C                                                                       00029590
CC                                                                      00029600
CC       USING THE PRINCIPLE OF SUPERPOSITION                           00029610
CC                                                                      00029620
C                                                                       00029630
   20 M = 0                                                             00029640
      IF  (LOAD. NE . 0)    M = M + 1                                   00029650
      IF  (PRES .NE.0.DO)    M = M + 1                                  00029660
      IF  (OMEGA.NE.0.DO)    M = M + 1                                  00029670
      IF  (M. EQ . 0)    GO  TO  40                                     00029680
      DO  41  I = 1 , IVEC                                              00029690
      DO  41  J = 1 , NDF                                               00029700
      DO  41  K = 1 , M                                                 00029710
      F(I,J) = F(I,J) + F(IVEC+K,J)                                     00029720
   41 CONTINUE                                                          00029730
C                                                                       00029740
CC                                                                      00029750
   40 CONTINUE                                                          00029760
      RETURN                                                            00029770
      END                                                               00029780
```

```
C                                                                       00029790
C
      SUBROUTINE  STRESS                                                00029800
                                                                        00029810
C                                                                       00029820
CCCCCCCCCC*************************************************************00029830
CCCCCCCCCC                                                            *00029840
CCCCCCCCCC  AT THE TWO GAUSS POINTS ON THE TWO BOUNDARIES OF          **00029850
CCCCCCCCCC  EACH ELEMENT (WHERE ETA=+1 AND ETA=-1) STRESSES           **00029860
CCCCCCCCCC  ARE CALCULATED IN THIS SUBROUTINE, AND ALSO THE           **00029870
CCCCCCCCCC  MAXIMUM 'MEAN AND OCTAHEDRAL' SHEARING STRESSES           **00029880
CCCCCCCCCC  ARE EVALUATED AND THEN THE TIME AND LOCATION OF           **00029890
CCCCCCCCCC  THE MAXIMUM STRESSES ARE DISTINGUISHED HERE.              **00029900
CCCCCCCCCC                                                            *00029910
CCCCCCCCCC*************************************************************00029920
C                                                                       00029930
C                                                                       00029940
      IMPLICIT  REAL * 8 (A-H,O-Z)                                      00029950
C                                                                       00029960
      COMMON /ONE/ AJ(2,2),B(4,16),COPY(37,298),D(4,4),DL(4,4),         00029970
     1DNZR(2,8),ESM(16,16),F(24,298),SSM(298,66),STOR(20,150)           00029980
C                                                                       00029990
      COMMON /TWO/ AL(5),DENS(5),DETA(8),DXI(8),E(5),EPSO(4),           00030000
     1FE(16),GC(4),GX(4),POI(5),R(149),SHP(8),SHT(5),TK(5),Z(149)       00030010
C                                                                       00030020
      COMMON /TRE/ AXIALF,CTI,EVERY,FORS1,OMEGA,HTCI,HTCO,              00030030
     1POSI,PRES,Q1,Q2,Q3,SHMAX,SMAX,TIME,TIME1,TINIT,TMAXM,TREF,        00030040
     2TMAXO,RMAXM,RMAXO,ZMAXM,ZMAXO,Q4,BIG                              00030050
C                                                                       00030060
      COMMON /FOR/ NN(40,9)                                             00030070
C                                                                       00030080
      COMMON /FIV/ NCFI(37),NCFO(37),NNR(37),NRE(37),NNL(37),           00030090
     1NPN(37)                                                           00030100
C                                                                       00030110
      COMMON /SIX/ IBAND,IEXT,INTP,IVEC,JVEC,KK,LOAD,NCFIT,NCFOT,       00030120
     1NCHECK,NET,NDF,NGP,NNE,NNLT,NNRT,NPASS,NPROB,NRPPI,NRPPO,         00030130
     3NNT,NNRE,NPNT,IPLANE,NBAND                                        00030140
C                                                                       00030150
      DIMENSION  EDISP(16),SIG(4),EPS(4)                                00030160
C                                                                       00030170
      GX(1) = -.577350269189626                                         00030180
      GX(2)=-GX(1)                                                      00030190
      GC(1)=1.0D0                                                       00030200
      GC(2)=GC(1)                                                       00030210
      PAI = 3.141592653589793                                           00030220
      NNE2 = 2 * NNE                                                    00030230
      DO  25  LL= 1 ,NET                                                00030240
```

```
C                                                                       00030250
      WRITE  (6,200)  LL                                                00030260
  200 FORMAT  (1H1,//,40X,'STRESSES AND TEMPERATURES IN ELEMENT',       00030270
     1I3,/,40X,39('*'),/,40X,39('*'),/)                                 00030280
C                                                                       00030290
      L = LL                                                            00030300
C                                                                       00030310
      IF  (L. EQ .1 ). GO  TO  1                                        00030320
      IF  (NN(L,9).EQ.NN(L-1,9))  GO  TO  3                             00030330
C                                                                       00030340
    1 K = NN(L,9)                                                       00030350
      EL = 1.D0 - POI(K)                                                00030360
      FACT = (E(K)*EL)/((1.D0+POI(K))*(1.D0-2.D0*POI(K)))               00030370
      EL = POI(K) / EL                                                  00030380
C                                                                       00030390
      D(1,1) = FACT                                                     00030400
      D(1,2) = EL * FACT                                                00030410
      D(1,3) = D(1,2)                                                   00030420
      D(1,4) = 0.D0                                                     00030430
      D(2,2) = FACT                                                     00030440
      D(2,3) = D(1,2)                                                   00030450
      D(2,4) = 0.D0                                                     00030460
      D(3,3) = FACT                                                     00030470
C                                                                       00030480
                                                                        00030490
      D(3,4) = 0.D0                                                     00030500
      D(4,4) = E(K)/(2.D0*(1.D0+POI(K)))                                00030510
C                                                                       00030520
C                                                                       00030530
      DO  3  I = 1,4                                                    00030540
      DO  3  J = I,4                                                    00030550
      D(J,I) = D(I,J)                                                   00030560
C                                                                       00030570
    3 CONTINUE                                                          00030580
C                                                                       00030590
C                                                                       00030600
CC                                                                      00030610
C                                                                       00030620
      ETA = -1.D0                                                       00030630
      DO  9  N = 1 , 2                                                  00030640
      ETA = - ETA                                                       00030650
      DO  13  M = 1 , 2                                                 00030660
      XI = GX(M)                                                        00030670
C                                                                       00030680
C                                                                       00030690
      DO  7  I = 1,4                                                    00030700
      DO  7  J = 1,NNE2                                                 00030710
    7 B(I,J) = 0.D0                                                     00030720
```

```
C                                                                       00030730
C                                                                       00030740
C                                                                       00030750
      SHP(1) = .25D0 * (1.D0-XI) * (1.D0-ETA) * (-XI-ETA-1.D0)          00030760
      SHP(2) = .5D0 * (1.D0-XI**2) * (1.D0-ETA)                         00030770
      SHP(3) = .25D0 * (1.D0+XI) * (1.D0-ETA) * (XI-ETA-1.D0)           00030780
      SHP(4) = .5D0 * (1.D0-ETA**2) * (1.D0+XI)                         00030790
      SHP(5) = .25D0 * (1.D0+XI) * (1.D0+ETA) * (XI+ETA-1.D0)           00030800
      SHP(6) = .5D0 * (1.D0-XI**2) * (1.D0+ETA)                         00030810
      SHP(7) = .25D0 * (1.D0-XI) * (1.D0+ETA) * (-XI+ETA-1.D0)          00030820
      SHP(8) = .5D0 * (1.D0-ETA**2) * (1.D0-XI)                         00030830
C                                                                       00030840
CCCCC                                                                   00030850
CCCCC  DERIVATIVES OF THE SHAPE FUNCTIONS WITH RESPECT TO XI.           00030860
CCCCC                                                                   00030870
C                                                                       00030880
      DXI(1)     = .25D0 * (1.D0-ETA) * (2.D0*XI+ETA)                   00030890
      DXI(2)     = -XI * (1.D0-ETA)                                     00030900
      DXI(3)     = .25D0 * (1.D0-ETA) * (2.D0*XI-ETA)                   00030910
      DXI(4)     = .5D0 * (1.D0-ETA**2)                                 00030920
      DXI(5)     = .25D0 * (1.D0+ETA) * (2.D0*XI+ETA)                   00030930
      DXI(6)     = -XI * (1.D0+ETA)                                     00030940
      DXI(7)     = .25D0 * (1.D0+ETA) * (2.D0*XI-ETA)                   00030950
      DXI(8)     = -.5D0 * (1.D0-ETA**2)                                00030960
C                                                                       00030970
CCCCC                                                                   00030980
CCCCC  DERIVATIVES OF THE SHAPE FUNCTIONS WITH RESPECT TO ETA.          00030990
CCCCC                                                                   00031000
C                                                                       00031010
      DETA(1)    = .25D0 * (1.D0-XI) * (2.D0*ETA+XI)                    00031020
      DETA(2)    = -.5D0 * (1.D0-XI**2)                                 00031030
      DETA(3)    = .25D0 * (1.D0+XI) * (2.D0*ETA-XI)                    00031040
      DETA(4)    = -ETA * (1.D0+XI)                                     00031050
      DETA(5)    = .25D0 * (1.D0+XI) * (2.D0*ETA+XI)                    00031060
      DETA(6)    = .5D0 * (1.D0-XI**2)                                  00031070
      DETA(7)    = .25D0 * (1.D0-XI) * (2.D0*ETA-XI)                    00031080
      DETA(8)    = -ETA * (1.D0-XI)                                     00031090
C                                                                       00031100
CCCCC                                                                   00031110
CCCCC     CONSTRUCTION OF THE JACOBIAN MATRIX.                          00031120
CCCCC                                                                   00031130
C                                                                       00031140
      DO  20  I = 1 , 2                                                 00031150
      DO  20  J = 1 , 2                                                 00031160
   20 AJ(I,J) = 0.D0                                                    00031170
C                                                                       00031180
C                                                                       00031190
      DO  22  I = 1 , NNE                                               00031200
```

```
      AJ(1,1) = AJ(1,1) + DXI(I)   * Z(NN(L,I))                          00031210
      AJ(1,2) = AJ(1,2) + DXI(I)   * R(NN(L,I))                          00031220
      AJ(2,1) = AJ(2,1) + DETA(I) * Z(NN(L,I))                           00031230
      AJ(2,2) = AJ(2,2) + DETA(I) * R(NN(L,I))                           00031240
   22 CONTINUE                                                           00031250
C                                                                        00031260
C                                                                        00031270
      DET =AJ(1,1) * AJ(2,2) - AJ(1,2) * AJ(2,1)                         00031280
      DEI =1.0D0/DET                                                     00031290
C                                                                        00031300
CC                                                                       00031310
CC    INVERTING THE JACOBIAN AND STORING IN  AJ  AGAIN.                  00031320
CC                                                                       00031330
C                                                                        00031340
      DUM = AJ(1,1) * DEI                                                00031350
      AJ(1,1) = AJ(2,2) * DEI                                            00031360
      AJ(1,2) = -AJ(1,2) * DEI                                           00031370
      AJ(2,1) = -AJ(2,1) * DEI                                           00031380
      AJ(2,2) = DUM                                                      00031390
C                                                                        00031400
CC                                                                       00031410
C                                                                        00031420
      CR = 0.D0                                                          00031430
      DO  2  I = 1,NNE                                                   00031440
      M1 = NN(L,I)                                                       00031450
    2 CR = CR + SHP(I) * R(M1)                                           00031460
C                                                                        00031470
CC                                                                       00031480
CC                                                                       00031490
CC    EVALUATION OF THE B MATRIX                                         00031500
CC                                                                       00031510
C                                                                        00031520
      DO  15  I = 1, NNE                                                 00031530
      K1 = 2 * I - 1                                                     00031540
      K2 = K1 + 1                                                        00031550
      B(1,K2) = AJ(1,1) * DXI(I) + AJ(1,2) * DETA(I)                     00031560
      B(2,K1) = AJ(2,1) * DXI(I) + AJ(2,2) * DETA(I)                     00031570
      B(3,K1) = SHP(I) / CR                                              00031580
      B(4,K1) = B(1,K2)                                                  00031590
      B(4,K2) = B(2,K1)                                                  00031600
   15 CONTINUE                                                           00031610
C                                                                        00031620
CC       POSITION OF GAUSS POINT                                         00031630
CC                                                                       00031640
C                                                                        00031650
      RPOS = 0.D0                                                        00031660
      ZPOS = 0.D0                                                        00031670
      DO  8  I = 1 , NNE                                                 00031680
```

```
      RPOS = RPOS + SHP(I) * R(NN(L,I))                                 00031690
    8 ZPOS = ZPOS + SHP(I) * Z(NN(L,I))                                 00031700
C                                                                       00031710
CC                                                                      00031720
      WRITE  (6,201)  XI , ETA , RPOS , ZPOS                            00031730
  201 FORMAT  (//,40X,'AT LOCATION XI=',1PD14.7,'  AND ETA=',           00031740
     11PD14.7,/,53X,'R=',1PD14.7,'  AND  Z =',1PD14.7,//,4X,            00031750
     2'TIME',8X,'SIGMA  Z',9X,'SIGMA  R',8X,'SIGMA THETA',7X,           00031760
     3'TAU  R Z',7X,'TEMPERATURE',6X,'MEAN STRESS',7X,'OCTA ',          00031770
     4'SHEAR',//)                                                       00031780
C                                                                       00031790
      DO  30  KK = 1 , IVEC                                             00031800
C                                                                       00031810
CCCCC                                                                   00031820
         EVALUATION OF THE DISPLACEMENT OF ELEMENT L                    00031830
CCCCC                                                                   00031840
      DO  4  I = 1 , NNE                                                00031850
      I2 = I * 2                                                        00031860
      I1 = I2 - 1                                                       00031870
      J2 = NN(L,I) *                                                    00031880
      J1 = J2 - 1                                                       00031890
      EDISP(I1) = F(KK,J1)                                              00031900
    4 EDISP(I2) = F(KK,J2)                                              00031910
C                                                                       00031920
CCCCC                                                                   00031930
         EVALUATING TEMPERATURE AT GAUSS POINT                          00031940
CCCCC                                                                   00031950
      TGP = 0.DO                                                        00031960
      DO  31  I = 1 , NNE                                               00031970
      M1 = NN(L,I)                                                      00031980
   31 TGP = TGP + SHP(I) * STOR(KK,M1)                                  00031990
C                                                                       00032000
      TGP = TGP -TREF                                                   00032010
      TGP = AL(NN(L ,9)) * TGP                                          00032020
C                                                                       00032030
CCCCCC                                                                  00032040
         EVALUATION OF STRAIN VECTOR FOR A GAUSS POINT ON               00032050
         THE BOUNDARY OF ELEMENT L .                                    00032060
CCCCCC                                                                  00032070
      DO  5  I = 1 , 4                                                  00032080
      G = 0.DO                                                          00032090
      DO  5  J = 1 , NNE2                                               00032100
      G = G + B(I,J) * EDISP(J)                                         00032110
    5 EPS(I) = G                                                        00032120
C                                                                       00032130
                                                                        00032140
                                                                        00032150
                                                                        00032160
```

```
C          NOTE: INITIAL STRAIN (4) IS ZERO AND IS NOT                    00032170
C          BEING SUBTRACTED FROM EPS(4)                                   00032180
C                                                                         00032190
      DO  6  I = 1 , 3                                                    00032200
    6 EPS(I) = EPS(I) - TGP                                               00032210
C                                                                         00032220
C                                                                         00032230
      DO  10  I = 1 , 4                                                   00032240
      G = 0.D0                                                            00032250
      DO  10  J = 1 , 4                                                   00032260
      G = G + D(I,J) * EPS(J)                                             00032270
   10 SIG(I) = G                                                          00032280
C                                                                         00032290
CC                                                                        00032300
CC                                                                        00032310
CC                                                                        00032320
CC         SMEAN  IS THE MEAN STRESS                                      00032330
CC         TMAXM  IS THE TIME WHEN MAX. MEAN STRESS OCCURS.               00032340
CC         OCTA   IS THE OCTAHEDRAL SHEAR STRESS                          00032350
CC         TMAXO  IS THE TIME WHEN MAX. OCTAHEDRAL SHEAR                  00032360
CC         STRESS OCCURS.                                                 00032370
C                                                                         00032380
      SMEAN = (SIG(1)+SIG(2)+SIG(3))/3.D0                                 00032390
C                                                                         00032400
CCC                                                                       00032410
CCC                                                                       00032420
CCC        CALCULATION OF OCTAHEDRAL SHEAR STRESS                         00032430
C                                                                         00032440
      OCTA = 0.D0                                                         00032450
      OCTA = OCTA + (SIG(1)-SIG(2))**2 + (SIG(2)-SIG(3))**2               00032460
      OCTA = (OCTA + (SIG(3)-SIG(1))**2) / 9.D0                           00032470
      OCTA = OCTA + (2.D0/3.D0) * SIG(4)**2                               00032480
      OCTA = DSQRT(OCTA)                                                  00032490
C                                                                         00032500
CC                                                                        00032510
C                                                                         00032520
      TGP = TGP / AL(NN(L,9)) + TREF                                      00032530
      TIME = STOR(KK,NNT+1)                                               00032540
C                                                                         00032550
      WRITE  (6,202)  (TIME,(SIG(I),I=1,4),TGP,SMEAN,OCTA)               00032560
  202 FORMAT  (3X,F7.2,7(3X,1PD14.6))                                     00032570
C                                                                         00032580
CC                                                                        00032590
CC                                                                        00032600
CC         FINDING MAX. MEAN STRESS                                       00032610
C                                                                         00032620
      SMEAN = DABS(SMEAN)                                                 00032630
      IF  (SMEAN.GT.SMAX)  GO  TO  27                                     00032640
```

```
      GO  TO  29                                                        00032650
   27 SMAX = SMEAN                                                      00032660
      TMAXM = STOR(KK,NNT+1)                                            00032670
      RMAXM = RPOS                                                      00032680
      ZMAXM = ZPOS                                                      00032690
   29 CONTINUE                                                          00032700
C                                                                       00032710
CCC                                                                     00032720
CCC        FINDING MAX. OCTAHEDRAL SHEAR STRESS.                        00032730
CC                                                                      00032740
C                                                                       00032750
      IF  (OCTA.GT.SHMAX)  GO  TO  28                                   00032760
      GO  TO  30                                                        00032770
   28 SHMAX = OCTA                                                      00032780
      TMAXO = STOR(KK,NNT+1)                                            00032790
      RMAXO = RPOS                                                      00032800
      ZMAXO = ZPOS                                                      00032810
C                                                                       00032820
CCC                                                                     00032830
C                                                                       00032840
   30 CONTINUE                                                          00032850
   13 CONTINUE                                                          00032860
    9 CONTINUE                                                          00032870
   25 CONTINUE                                                          00032880
C                                                                       00032890
CC                                                                      00032900
      RETURN                                                            00032910
      END                                                               00032920
```

APPENDIX C

USER'S MANUAL

In this section the procedure for the use of the present computer program is described. It is intended that a person with minimum familiarity with the details of the finite element method and computer programming be able to use this program.

The steps below are in order and the user is advised to follow them carefully.

For finding the thermal stresses in an axisymmetric body under axisymmetric loading, the user has the choice of using either the Metric or British system of units. The program will handle both systems and the necessary conversions are made automatically within the program. However, the units used in each system must be consistent and they should be as listed below in Table V.

Now for preparation of the data input for a given axisymmetric geometry with prescribed thermal and structural boundary conditions we go through the details with a simple example.

Step 1:

Draw the longitudinal cross-section of the body to scale. Identify the cylindrical coordinates R and Z, with the origin of the Z axis on the most left-hand point of the

TABLE V

UNITS FOR INPUT DATA

Note: an input card specifies whether British or Metric data is being used.

Quantity	British Unit Syst.	Metric Syst.
Coordinates	in.	cm
Time	sec.	sec.
Temperature	°F	°C
Velocity	ft./sec.	m/sec.
Axial force[✓]	lbf	kg
Pressure	lbf/in^2	kg/cm^2
Density	lbm/ft^3	gm/cm^3
Specific heat	Btu/lbm°F	cal/kg°C
Mod. of elasticity	lbf/in^2	kg/mm^2
Coef. of thermal exp.	1/°F	1/°C
Thermal conductivity	Btu/hr.ft.°F	cal/sec.cm.°C
Heat transfer coef.	$Btu/hr.ft.^2°F$	$cal/sec.cm.^2°C$
Load vector	lbf	kg
Rotational speed	Revolutions/min.	Revolutions/min.

[✓] lbf is pound force and lbm is pound mass.

body. The entire cross section must lie in the first quadrant of the coordinate plane.

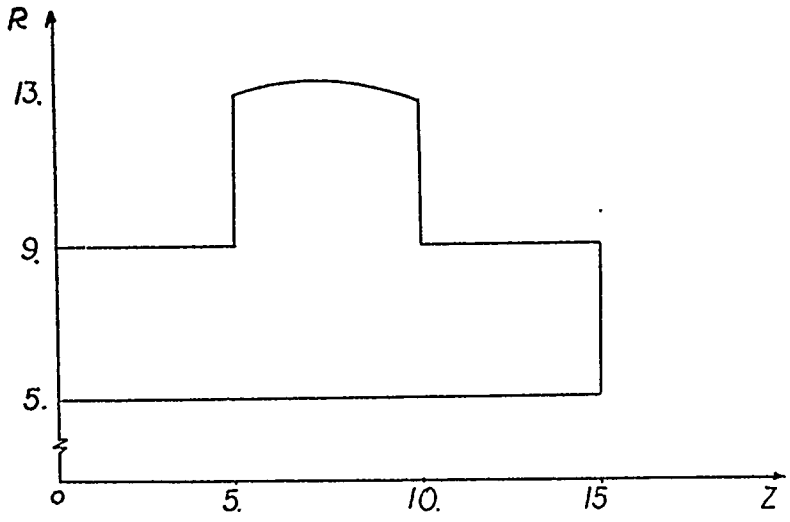


Figure 14. Longitudinal Cross Section.

Step 2:

Subdivide the cross section into a maximum of 40 quadrilateral elements. This subdivision is extremely important and we should provide smaller elements for the parts of the cross section where we expect the largest stresses or temperature gradient. If the body is made from several materials, elements should be chosen so that each element contains only one material. Any two adjacent elements must share one complete side of the quadrilateral. Number the elements, starting from 1, in any arbitrary manner (see Fig. 15).

Step 3:

Since eight-noded elements are used in the program, identify these nodal points around the boundary of each element. Four nodes will be at the corners and the other four will be

at the mid-points of the sides. Number sequentially these nodes starting from 1 and increasing in the direction where the number of elements is the least. (The numbers thus assigned are called global node numbers.) For clarity of this step assume a cross-section as in Figure 15 such that the maximum number of elements in one direction is less than the maximum number of elements in other direction. (In Fig. 15 we have maxima of 2 and 3 elements in R and Z directions respectively.) Thus we number the nodes beginning in the R direction. See Fig. 16. This method will give the minimum band width of the stiffness matrix and will save computer execution time.

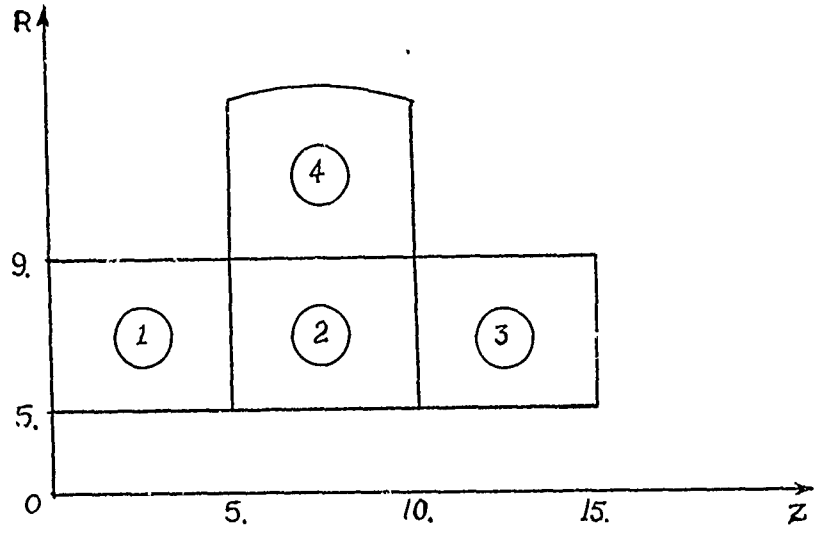


Figure 15. Subdivision Inot Elements.

Step 4:

At this point we are ready to prepare the first data card. Input quantities are:

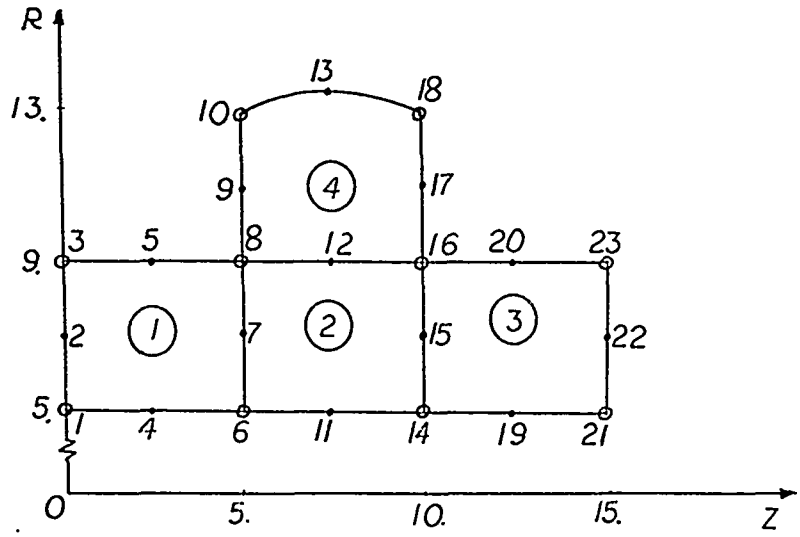


Figure 16. Element and Node Numbering.

NET	The total number of elements
NNT	Total number of nodes
NCN	Total number of corner nodes
NØFN	Total number of mid-side nodes not lying on the straight line joining the corner nodes (number of "Off" nodes)
NMAT	Number of different materials
NPRØB	Number of problems to be solved

For the present we take NPRØB = 1.

Prepare a single card that reads

NET, NNT, NCN, NØFN, NMAT, NPRØB

with the format (8I5). For our example, if all the elements are from the same material this data card reads:

```
    4   23   10    1    1    1
```

On this card, or any succeeding one, if an input quantity (e.g., NØFN) is zero, the corresponding field may have a 0 or be left blank, unless otherwise stated.

Step 5:

For each corner node a card should be punched which reads: The node number I, the R coordinate and the Z coordinate of the node. The format is (I5, 2F10.5).

It is not necessary to sort these cards in the order of increasing or decreasing corner node numbers, they can be put together in any arbitrary order. A typical card is illustrated in step 7. There must be as many punched cards as the number of corner nodes (NCN).

Step 6:

In this step we read in the connectivity array, i.e., for each element we prepare one card which gives the global node numbers and the material identification number.

For each element, start from the lower left-hand node and move in the counterclockwise direction within that element and punch the global node numbers in order. The format is (9I5) where the last I5 is the material identification number.

It is very important that the cards prepared for this step be put together in order of elements, i.e., the first card for element 1, the second card for element 2, etc. For ease of sorting the connectivity cards the element number may be punched after column 50. As an example, for element 2 of Fig. 16 the connectivity card should read:

```
      6   11   14   15   16   12    8    7    1    2   13
```

```
0000000000000000000000000000000000000000000000000000000000000000000000000000001
1234567890123456789012345678901234567890123456789012345678901234567890123456789
```

where 1 identifies the type of material in this element and 2 in column 55 stands for the element number.

Step 7:

If each side of every element is straight, NØFN = 0 and this step is omitted.

For each mid-side node that is on a curved element edge a card is prepared with format (I5,2F10.5) to read the global node number and the R and Z coordinates. Here, as in step 5, these cards may be put together in any arbitrary order. For the case of Fig.16, there will be only one card to be punched and it would read:

```
   13  13.75D0   7.5D0
```

```
0000000000000 0000000 0000000000000000000000000000000000000000000000000000000001
1234567890123 4567890 1234567890123456789012345678901234567890123456789012345678
```

To assure in all cases that the double precision capability of the program is not compromised by less precise floating point inputs, it is recommended that all such inputs be made in D-FORMAT. The F-FORMAT instructions merely allocate input card fields.

Step 8:

The properties of the different materials are specified in this step. For each material a card must be prepared that gives the modulus of elasticity (E), coefficient of thermal expansion (AL), Poisson's ratio (PØI), thermal conductivity (TK), density (DENS) and specific heat (SHT). I.e., read:

E, AL, PØI, TK, DENS, SHT

with the format (2D12.4, 4F8.3). The first card will be for material number 1, the second card for material number 2, etc.

Step 9:

For this step a single card must be punched, starting in column one, which reads the word BRITISH or METRIC in accordance with the system of units used.

This completes the input of geometric and material property data.

Step 10:

For a transient temperature problem only (no stress calculations) leave a blank card for this step and proceed directly to step 16.

For a stress problem or thermal stress problem, in this step we specify the type of problem and the structural boundary conditions.

The following quantities, when pertinent, are to be specified.

ØMEGA The speed of rotation, about the Z axis, in revolutions per minute

PRES The external pressure applied to a boundary segment

IEXT The number of known temperature vectors for which evaluation of thermal stresses is desired

LØAD The indication for an additional known load vector. If there is one, LØAD = 1, otherwise LØAD = 0

NNLT Total number of nodes fixed in the longitudinal direction

NNRT Total number of nodes fixed in the radial direction

IPLANE An indication for the plane-end boundary condition. If desired IPLANE = 1

If any of these items is not applicable, the corresponding field is left blank.

Now we have to punch a single card for this step which reads:

ØMEGA, PRES, IEXT, LØAD, NNLT, NNRT, IPLANE

with format (2F10.4, 5I5).

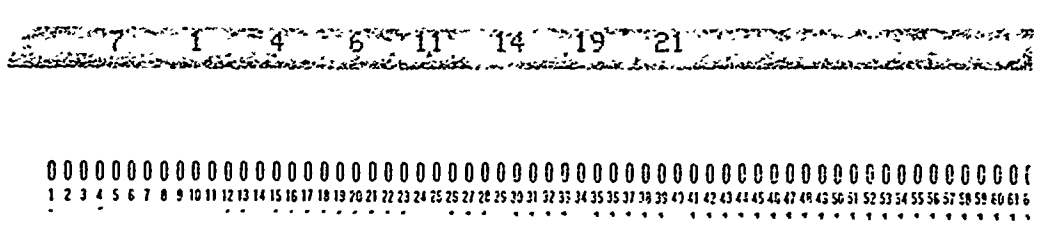
Step 11:

If there is no pressure loading (PRES=0), omit this step. For non-zero pressure, the total number of nodes on pressure loading boundary segment (NPNT) and the global node numbers of these (pressure) nodes (NPN(I)) must be specified here. Only one segment is permitted.
We read in:

NPNT, (NPN(I),I=1,NPNT)

with the format (12I5).

In our example, Fig. 16, if there is pressure inside the body, then the data card for this step will be:



Step 12:

If there is no additional load vector (LØAD = 0), omit this step.

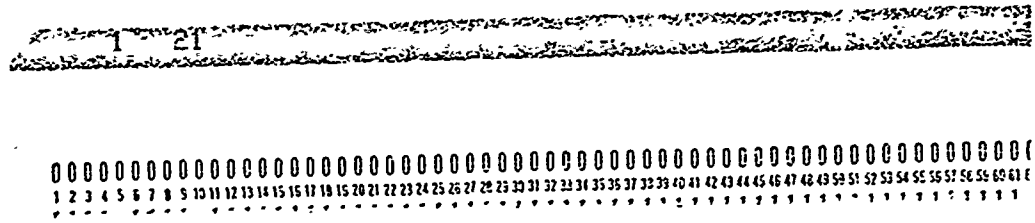
For LØAD = 1 read in the additional load vector with the format (6F10.4). The components of the load vector are arranged in the order: R component at node 1, Z component at node 1, R component at node 2, etc. I.e., READ (F(I),I=1,NDF), where NDF (the number of degrees of freedom) is equal to two times the total number of nodes (2*NNT).

Step 13:

In any stress problem, or thermal stress problem, there must be at least one node constrained against longitudinal motion, i.e., NNLT > 0, so we must specify here which nodes are to be constrained against longitudinal motion. These global node numbers NNL(I) are read in with the format (I0I5), i.e.,

READ: (NN(I),I=1,NNLT)

In our example, if global nodes 1 and 21 are fixed in the Z direction, we have a card that reads:



Step 14:

If there are no nodes fixed in the radial direction (NNRT = 0), omit this step.

For NNRT > 0 we must read in the global numbers (NNR(I)) of those nodes which are to be fixed against radial motion. The format is (10I5), i.e.,

READ: (NNR(I),I=1,NNRT)

If node 1 and 21 of our example are also fixed in the R direction, then the card for this step would read exactly as the one in Step 13.

Step 15:

If there is no plane-end boundary condition (IPLANE = 0), omit this step.

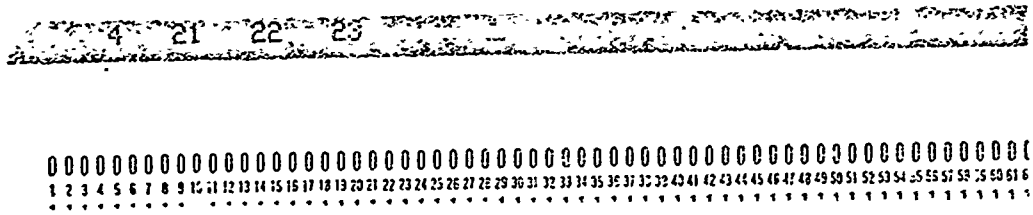
For the case of end-planes-remain-plane boundary condition, i.e., (IPLANE = 1), we have to specify the total number of nodes of the right-hand end (NNRE) and the global node numbers of this end (NNR(I)).

We read in:

NNRE, (NNR(I),I=1,NNRE)

with the format (10I5).

For the case of Fig. 16, if it is desired to have end-planes-remain-plane boundary condition then the data for this step will be:



Step 16:

In this step we begin the specification of thermal boundary conditions. Additional details are prescribed in Steps 19 through 24.

Thermal boundary conditions are imposed along discrete segments of the boundary. Each such segment must begin and end at a corner node of a boundary element. We consider separately the imposition of the various kinds of thermal boundary conditions.

(a) Convection

Since the fluid temperature for convection is determined from a prescribed temperature-time history at entry section and a specified flow velocity (see Appendix D, Section 5), it is necessary that the terms "inside," i.e., adjacent to the symmetry axis, and "outside" have their ordinary meanings. Thus on the inside surface the convection boundary condition may be applied to a single (continuous) segment. A similar prescription may be employed for the outside portion. The entry section used for flow calculations is at the upstream end of corresponding segment.

(b) Constant temperature

To specify constant temperature on a portion of the boundary, the designator "inside" or "outside" may be used. Such a constant temperature portion may consist of several

discrete segments. If two different constant temperature portions are prescribed, one may be designated "inside" and the other "outside."

(c) Combinations of convection and constant temperature

The convection and constant temperature conditions can be used together, but the portion designated "inside" must have only one of these conditions prescribed and the same restriction applies to the "outside."

(d) Insulated

All portions of the boundary not included in the segments specifically identified as "inside" and "outside" are considered insulated.

The following items, when pertinent, are to be given as specified below:

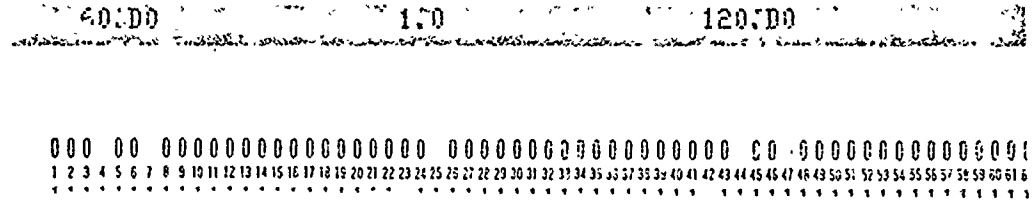
TINIT	The constant initial body temperature
Q2=0.	For insulated boundary condition outside
Q2 > 0.	For convection or constant temperature boundary condition outside
Q3=0.	For insulated boundary condition inside
Q3 > 0.	For convection or constant temperature boundary condition inside
Q4 < 0.	If solving stress problem only
Q4 > 0.	If solving temperature problem only
Q4=0.	If solving thermal stress problem
AXIALF	The axial force in the Z direction. As usual, + (plus) for tension and - (minus) for compression.

So, we read:

TINIT, Q2, Q3, Q4, AXIALF

with the format (5F10.4).

In case of Fig. 16, if the initial solid temperature is 60° and we have insulated outside and convection boundary condition inside with 120 kg axial compressive force, then the card for this step will be:



where the blanks left in the columns 31 through 40 indicate that we want to solve a thermal stress problem.

Step 17:

No action is taken in this step - we merely choose between proceeding to step 18 or jumping to step 26.

If there are no known temperature vectors for which evaluation of thermal stresses is desired, proceed to step 18.

For IEXT > 0, i.e., when some known temperature vectors are to be entered for thermal stress analysis alone or combined with some other loadings, proceed directly to step 26.

Step 18:

If (Q4 < 0); i.e., we are to solve only a stress problem, proceed directly to step 27.

Step 19:

If there is an insulated boundary condition outside (Q2 = 0), omit the following steps and start from step 22.

For the case of a convection or constant temperature boundary condition outside (Q2 > 0.), we have to specify the outside heat transfer coefficient (HTCØ)(for constant temperature HTCØ = 10^{20}.), the constant outside initial fluid temperature (TEMPØ) which may be equal to the solid's initial temperature, the total number of nodes in contact with fluid outside (NCFØT), and the number of temperature ramps for outside flow (NRAMPØ). We read:

HTCØ, TEMPØ, NCFØT, NRAMPØ

with the format (D16.4,F10.4,2I5). See example in step 22.

Step 20:

Here we read in the global node numbers of the nodes in contact with outside fluid (NCFØ(I)). The sequence of these node numbers must be in the direction of flow velocity. Read:

(NCFØ(I),I=1,NCFØT)

with the format (12I5) (see example in step 23).

Step 21:

For each ramp of outside flow we read in the time when the ramp starts BDRYØ(I,1), the initial temperature of the ramp BDRYØ(I,2), the final temperature of the ramp (specify only if it differs from the initial temperature of the next

ramp) BDRYØ(I,3) and the velocity of the fluid for this ramp BDRYØ(I,4), with the format (4F10.4) so we read:

((BDRYØ(I,J),J=1,4),I=1,NRAMPØ).

The number of cards prepared at this step is equal to the number of ramps. See example in step 24.

Step 22:

If there is an insulated boundary condition inside (Q3=0), proceed directly to step 25.

For the case of convection or constant temperature boundary condition inside, Q3 > 0. We have to specify the inside heat transfer coefficient (HTCI) (for constant inside temperature HTCI=10^{20}), the constant inside initial fluid temperature (TEMPI), which may be equal to the solid's initial temperature, total number of nodes in contact with fluid inside (NCFIT) and the number of ramps of the inside flow (NRAMPI). We read:

HTCI, TEMPI, NCFIT, NRAMPI

with the format (D16.4, F10.4,2I5).

For our example assume the time variation of entry temperature for the inside flow to be as in Fig. 17. The heat transfer coefficient is 175.0 with initial inside fluid temperature 70.0. The card for this step is:

```
175.D0      70.D0          7    3
```

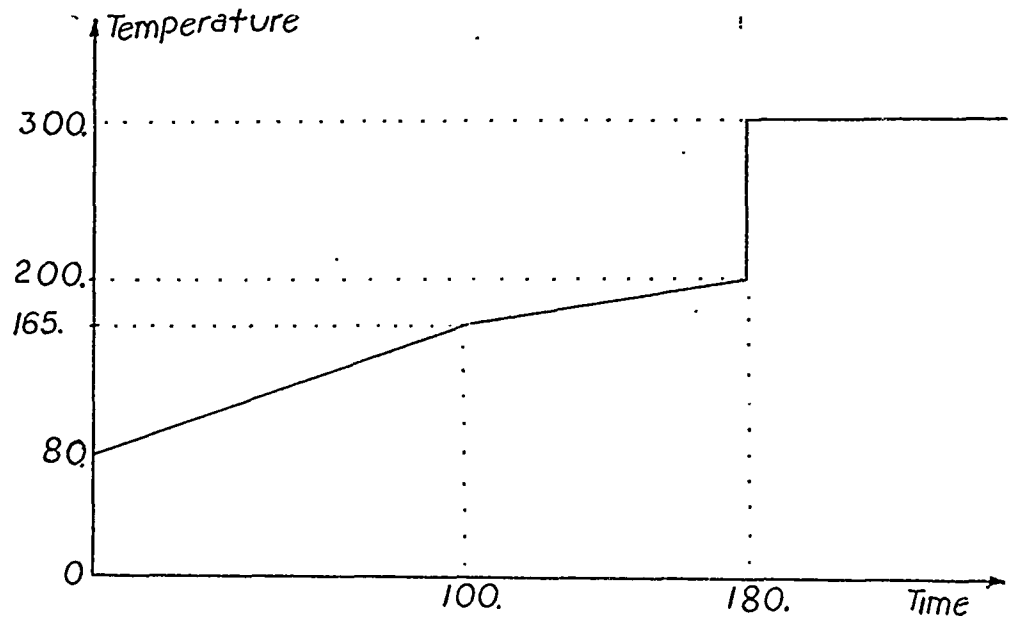


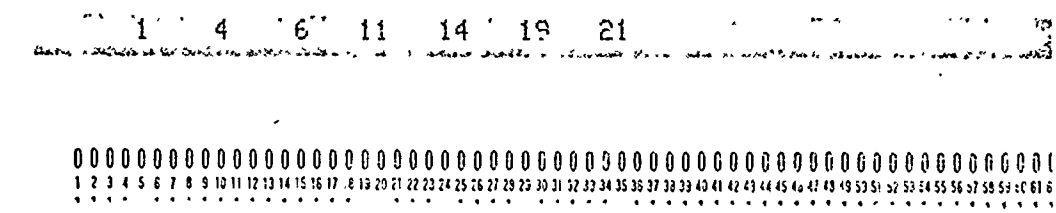
Figure 17. Entry Temperature-Time Variation.

Further details of this temperature-time history are specified in step 24.

Step 23:

Here we read in the global numbers of those nodes which are in contact with inside fluid, (NCFI(I)) with the format (12I5). It is important to read these node numbers in the direction of flow velocity.

For example, if the inside flow of Fig. 16 is from left to right, then for this step the card reads:



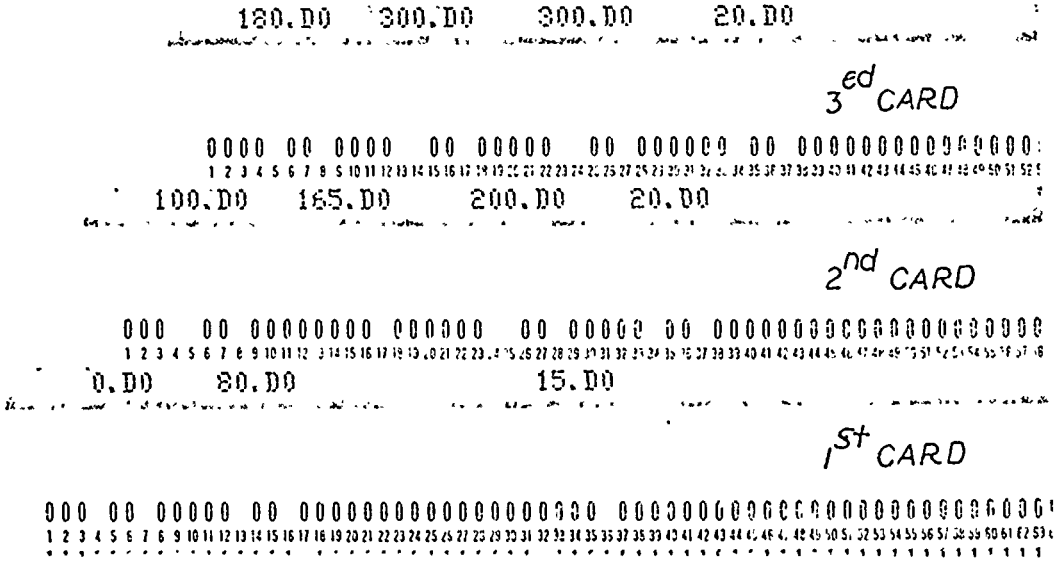
Step 24:

In this step data for inside flow are given. For each ramp I we read: the time when the ramp starts BDRYI(I,1), the initial temperature of ramp I BDRYI(I,2), the final temperature of ramp I (specify only if it differs from the initial temperature of the next ramp) BDRYI(I,3), and the velocity of the fluid for this ramp BDRYI(I,4). We read in:

((BDRYI(I,J),J=1,4),I=1,NRAMPI)

with format (4F10.4).

For example of Fig. 17, let the flow velocity for the first ramp be 15 and that for the second and third segment be 20. Three cards are needed:



Step 25:

In this step we must specify the following items:

DTI — The time increment (step size) of trapezoidal integration

TIME1 — The time when the first calculated temperature vector is to be stored for thermal stress evaluation

EVERY The interval of time between the successive temperature vectors which are to be stored

IVEC Total number of temperature vectors to be stored for thermal stress evaluation

INTP The indication for printing the calculated temperature vectors

If INTP = 0 there will be no temperature prints

If INTP = 1 the program will print the temperature after every step of time integration

If INTP = 5 the program will print the temperature after every 5 steps of time integrations, etc.

At this step we prepare a single card that reads:

DTI, TIME1, EVERY, IVEC, INTP

with format (3F10.4,2I5).

For example, if the step size of trapezoidal time integration is 2 sec. and we desire to evaluate thermal stresses for 10 different temperature vectors, each 40 seconds apart, and starting with the first thermal stress calculation at time equal to 60, then the card for this step is:

```
     2.D0      60.D0      40.D0       10   15
```

```
000000 000000 00 000000 00 0000000 0000000000000000000000000(
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 6
```

where 15 in columns 34 and 35 indicates that the temperatures will be printed every 15 steps of time integration. The total time of integration for this example is 420 seconds, since

TIME1 + EVERY * (IVEC-1) = 420.

Step 26:

If there are no known temperature vectors (IEXT = 0) for which thermal stress evaluation is desired, omit this step.

For IEXT > 0 we must specify the nodal temperatures which are to be stored for thermal stress evaluation. We read:

((STØR (I,J) , J=1,NNT), I=1,IEXT)

with the format (6F10.4).

This means that we enter all the components of the first nodal temperature vector, followed by the components of the second nodal temperature vector and so on until IEXT such vectors have been entered.

Step 27:

The data cards for this problem are completed now. If no more problems are to be solved for the same geometry, omit this step.

If another problem is to be solved for the same geometry and spatial discretization, increment by 1 the NPRØB in step 4 and start the input of the new problem from step 10.

Step 28:

If there is no access to the IBM 360 computer of the Naval Postgraduate School, omit the following steps and start from step 32.

For the convenience of the user, the author has put a so-called CHECK program and the program presented in

Appendix B in the computer system at N.P.S. (Naval Postgraduate School). It is advised, however, that the user check his input data deck with the CHECK program before attempting to solve the problem.

For using the CHECK program prepare the following control cards.

```
//XXXX0000 JØB (0000,0000FT,XX00),'NAME',TIME=1
//JØBLIB  DD  DSN=F0609.BAKH,DISP=S'iR,UNIT=2314,VØL=SER=DUFFY
//GØ  EXEC  PGM=CHECK,REGION=100K
//FT06F001  DD SYSØUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=3325),
//            UNIT=SYSØUT
//FT05F001  DD  *
```

where the first card is the regular FORTRAN job card used at this institution.

Prepare your deck as Fig. 18 and it may be read in from the hot card reader.

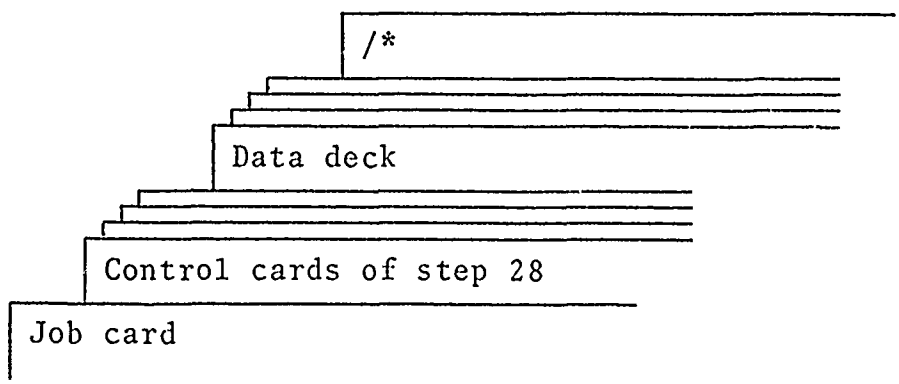


Figure 18. Set-up for Using CHECK Program.

If the output of the CHECK program has any error messages or has not been run, then there are some mistakes in the data deck or control cards.

If there are no error messages on the output of CHECK program, then study carefully the output and make sure it is the same problem that has been intended to be solved. Once the correctness of the input data has been insured, proceed to the next step.

Step 29:

AXITTS stands for Axisymmetric Transient Thermal Stress and this is the name given to the program presented in Appendix B when stored in the computer. For using that we must prepare the following control cards.

```
//XXXX0000 JØB (0000,000FT,XX00)'NAME',TIME=10
//JØBLIB  DD  DSN=F0609.BAKH,DISP=SHR,UNIT=2314,VØL=SER=DUFFY
//GØ  EXEC  PGM=AXITTS,REGION=425K
//FT06F001  DD SYSØUT=A,DCB=(RECFM=FBA,LRECL=133,BLKSIZE=3325),
//         UNIT=SYSØUT,SPACE=(CYL,(6,1))
//FT05F001  DD  *
```

where the first card is the regular FØRTRAN job card. It is advised to ask for 10 minutes time, i.e., TIME=10, since this would not affect the priority of the job within the class K jobs.

Prepare the deck as in Fig. 19 (it may be read into the computer from the hot card reader) and submit a so-called

service request card, available in the computer center, to the operator on duty.

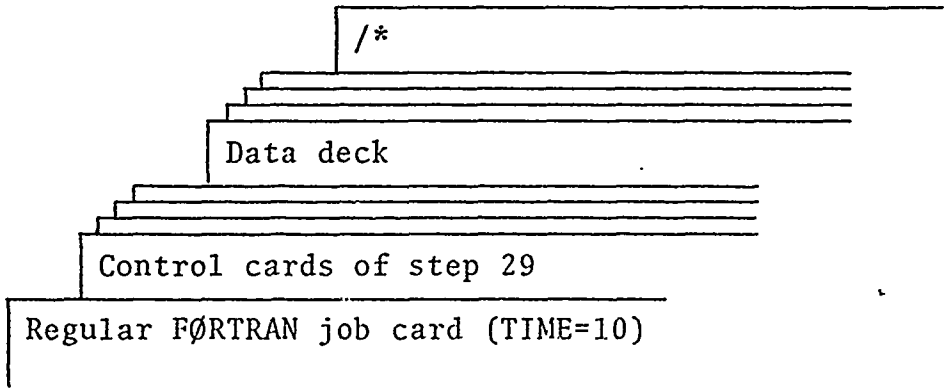


Figure 19. Deck Set-up for Using AXITTS Program.

Step 30:

If the user wishes to have a listing of the program he may prepare the following control cards as in Fig. 20 since the program is also listed on the data cell for this purpose.

```
Regular FØRTRAN job card
//PRNT  EXEC PGM=IEBPTCH
//SYSPRINT  DD DUMMY
//SYSUT1  DD DSN=F0609.BAKH,DISP=ØLD,UNIT=2321,
//    VØL=SER=CEL003,DCB=(RECFM=FB,LRECL=80,BLKSIZE=2000)
//SYSUT2  DD  SYSØUT=A,SPACE=(CYL,6)
//SYSIN  DD  *
       PRINT  TYPØRG=PØ,MAXFLDS=1,MAXNAME=1
       MEMBER  NAME=AXITTS
       RECØRD  FIELD=(80)
/*
```

Figure 20. Deck Set-up for Obtaining a Listing of the Program AXITTS.

The set-up deck of Fig. 20 may be read in from the hot card reader to get a listing of the program.

Step 31:

If the user wishes to obtain a deck of the program AXITTS he may prepare a deck as in Fig. 21 and read it in from the hot card reader. He must notify the operator on duty to be prepared for punching almost two boxes of IBM cards.

```
Regular FØRTRAN job card
//PUNCH  EXEC PGM=IEBPTCH
//SYSPRINT  DD DUMMY
//SYSUT1  DD DSN=F-0909.BAKH,DISP=ØLD,UNIT=2321,
//    VØL=SER=CEL003,DCB=(RECFM=FB,LRECL=80,BLKSIZE=2000)
//SYSUT2  DD  SYSØUT=B,SPACE=(CYL,6)
//SYSIN  DD  *
       PUNCH TYPØRG=PØ,MAXFLDS=1,MAXNAME=1
MEMBER  NAME=AXITTS
RECØRD  FIELD=(80)
/*
```

Figure 21. Deck Set-up for Obtaining a Punched Deck of the Program AXITTS

Step 32:

For using the program presented in Appendix B, if the user does not have access to the computer facility at N.P.S., he must punch a copy of the program. (Good luck!)

For the storage location 425K bytes are required and since the output may be longer than what usually is allowed, three additional cylinders are recommended. The execution time required depends on the size of the problem and the number of time integration steps; however, 10 minutes computer time would be sufficient for a problem of 140 nodes and 500 steps of time integration, with 20 temperature vectors stored for stress calculation.

APPENDIX D

PROGRAMMING

The computer program presented in Appendix B is formed from one main program and fifteen different subroutines. The communication between the main program and the subroutines is handled through the common blocks. In order to minimize storage requirements, most of the storage location of the system stiffness matrix is used for temperature calculations. This is accomplished by use of EQUIVALENCE statements. The total storage requirement is 425K bytes.

In this section the function of some parts of the program is discussed and the assumptions used are brought to attention.

1. MAIN PROGRAM

The main program is simply calling different subroutines when they are needed. The subroutines which are called in main program are:

INPUT, PRØB, CANDY, FLØW, FØRMV, TEMPER, PLØT, STIFF, FØRMF, CENTF, PRESS, DISPL, AND STRESS.

2. SUBROUTINE INPUT

In this subroutine the data regarding the geometry of the body, subdivisions into elements, and the material properties are read in and printed out. Calculation of the mid-side coordinates based on the straight line is done

here. This subroutine calls for subroutine PLØTRZ once to produce a graphical representation of the nodal points then the half-band-width of the system stiffness matrix is calculated from the connectivity array. Also, depending upon the choice of the system of units, the necessary conversions in each system of units are accomplished and the reference temperature (70°F or 20°C) based on the system of units is selected here.

3. SUBROUTINE PRØB

For each problem this subroutine is called once by the main program to read in and print out the information regarding the nature of the problem. Based upon the given information the type of the problem is distinguished here and for the thermal problems the length of time integration is calculated.

4. SUBROUTINE CANDY

Subroutine CANDY (C and Y) evaluates the capacitance matrix $\underline{\underline{C}}$ and admittance matrix $\underline{\underline{Y}}^{+}$ of Eq. 13, in banded form. For each problem this subroutine is called once. The functional flow chart of the subroutine CANDY is given in Fig. 22. Since the only non-zero elements of the symmetric matrix $\underline{\underline{Y}}^{*}$ (Eq. 11) are the diagonal, first and second super-diagonals, in order to conserve storage and reduce the number of arithmetic operations, the non-zero elements are stored in three different vectors (DİAG, ØFF1, ØFF2).

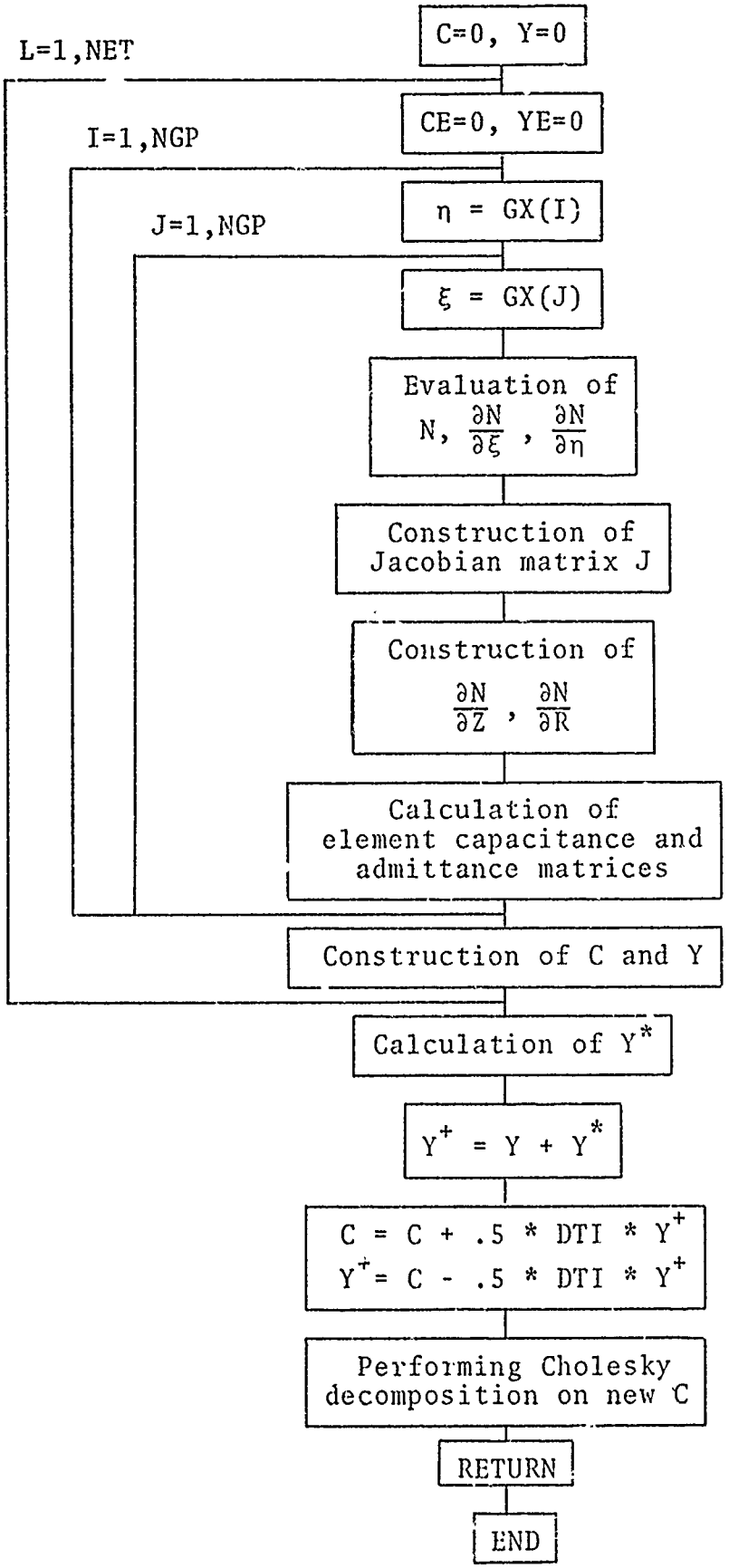


Figure 22. Functional Flow Chart of CANDY.

Once the system capacitance and admittance matrices $\underline{\underline{C}}$ and $\underline{\underline{Y}}^{+}$ are formed, then the matrices $\underline{\underline{A}}$ and $\underline{\underline{G}}$ (Eq. 29) are evaluated and replace $\underline{\underline{C}}$ and $\underline{\underline{Y}}$ respectively.

5. SUBRØUTINE FLØW

If there is any convection or constant temperature thermal boundary condition, this subroutine is called from the main program for each step of time integration in order to evaluate the temperature of the fluid nodes. The calculation is based upon the constant fluid flux assumption (for both inside and outside flow). Consider a section of an irregular cylindrical pipe as Fig. 23 and focus attention on element i+1 on the inner boundary.

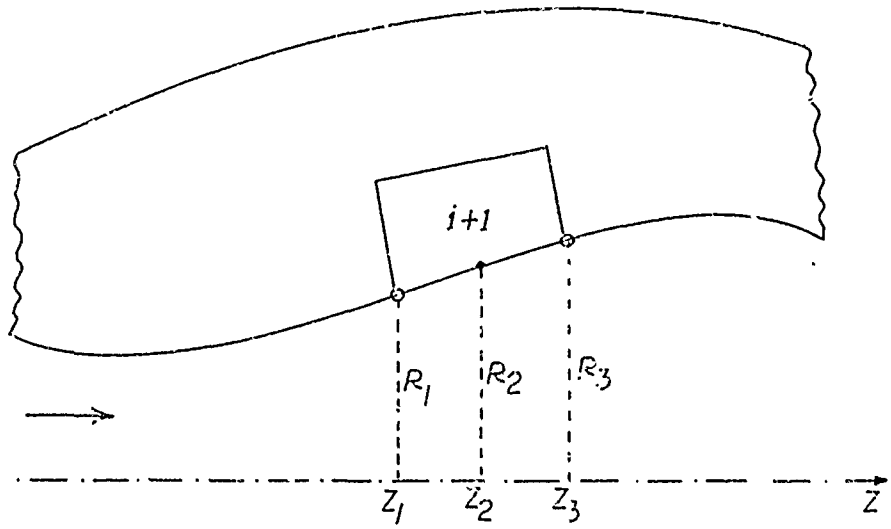


Figure 23. Fluid Node Representation for Inside Flow.

If fluid is moving from left to right we may number the R and Z coordinates of the fluid nodes of the element i+1 as in Fig. 23, then the volume of fluid pumped into the pipe at time τ is given by

$$Q_{in} = a\ v\ \tau, \tag{56}$$

where a and v are, respectively, the area and velocity at the entrance section. Also the volume of the pipe up to the mid-node of element i+1 is

$$Q_1 = V_i + \int_{Z_1}^{Z_2} \pi R^2\, dZ. \tag{57}$$

where V_i is the volume of the pipe between the entrance section and element i+1 on the inner boundary.

Now we can write

$$R = \sum_{i=1}^{3} R_i\ N_i, \tag{58}$$

where N_i are the one dimensional shape functions (given in Appendix A).

If we substitute Eq. 58 into 57 and make the coordinate transformation, after integration we get

$$Q_1 = V_i + \frac{\pi}{120}(Z_2 - Z_1)(31R_1^2 + 64R_2^2 + R_3^2 + 46R_1R_2 - 8R_1R_3 - 14R_2R_3). \tag{59}$$

Now by equating Eq. 59 to Eq. 56 at any time τ we may determine whether the front of the flow has already passed the mid-node of the wetted side of the element i+1.

A similar argument can be carried out for the end node of the element i+1. In that case we also get an equation similar to Eq. 59 with different numerical coefficients.

For the case of the outside flow it has also been assumed to have a constant fluid flux and a fictitious

entrance circular cross-sectional area (whose radius is the largest R plus unity) has been assumed.

The numerical coefficients of R's in Eq. 59 are used in subroutine FLØW and it has been assumed that the temperature of a fluid "particle" does not change during passage through the active section. This is believed to be an acceptable approximation for representative values of flow velocity and active length. With this assumption, for a given entry time-temperature relation (inside and outside flow) this subroutine evaluates the fluid nodal temperatures at any time.

6. SUBRØUTINE FØRMV

At every step of time integration this subroutine is called by the main program to form the vector $\underline{v}$ of the right-hand side of the discretized finite element Eq. 13) for the given thermal boundary conditions The program itself is self explanatory.

7. SUBRØUTINE TEMPER

The nodal temperatures are calculated in this subroutine with the trapezoidal time integrations. Irons' correction is applied here after every 10 steps of time integration. The temperature vectors selected for the stress analysis are stored. The transient temperatures will be printed out at the desired interval of time.

8. SUBRØUTINE STIFF

For any stress or thermal stress problem subroutine STIFF is called once by the main program to evaluate the stiffness matrix of the system. Once the stiffness matrix of the system is calculated then the desired structural boundary conditions are applied and, at the end, the Cholesky decomposition is performed. The functional flow chart of subroutine STIFF is given in Fig. 24.

9. FØRMF

In subroutine FØRMF the thermal load vectors are calculated for as many as (IVEC) given temperature vectors. These thermal load vectors are the columns of a rectangular matrix $\underline{\underline{F}}$. The provision is made that no matter how many temperature vectors are given (always IVEC $\leq$ 20) the IVEC number of columns of the $\underline{\underline{F}}$ are filled with the thermal load vectors and the last column of F will contain the load vector corresponding to the unit end displacement for zero axial force when the plane-end boundary condition is applied. The flow chart of subroutine FØRMF is given in Fig. 25.

10. SUBRØUTINE CENTF

If the system is rotating about the axis of revolution, this subroutine is called once by the main program to evaluate the centrifugal load vector. This load vector is always placed in the first column after the thermal load vectors (IVEC+1 position) in $\underline{\underline{F}}$.

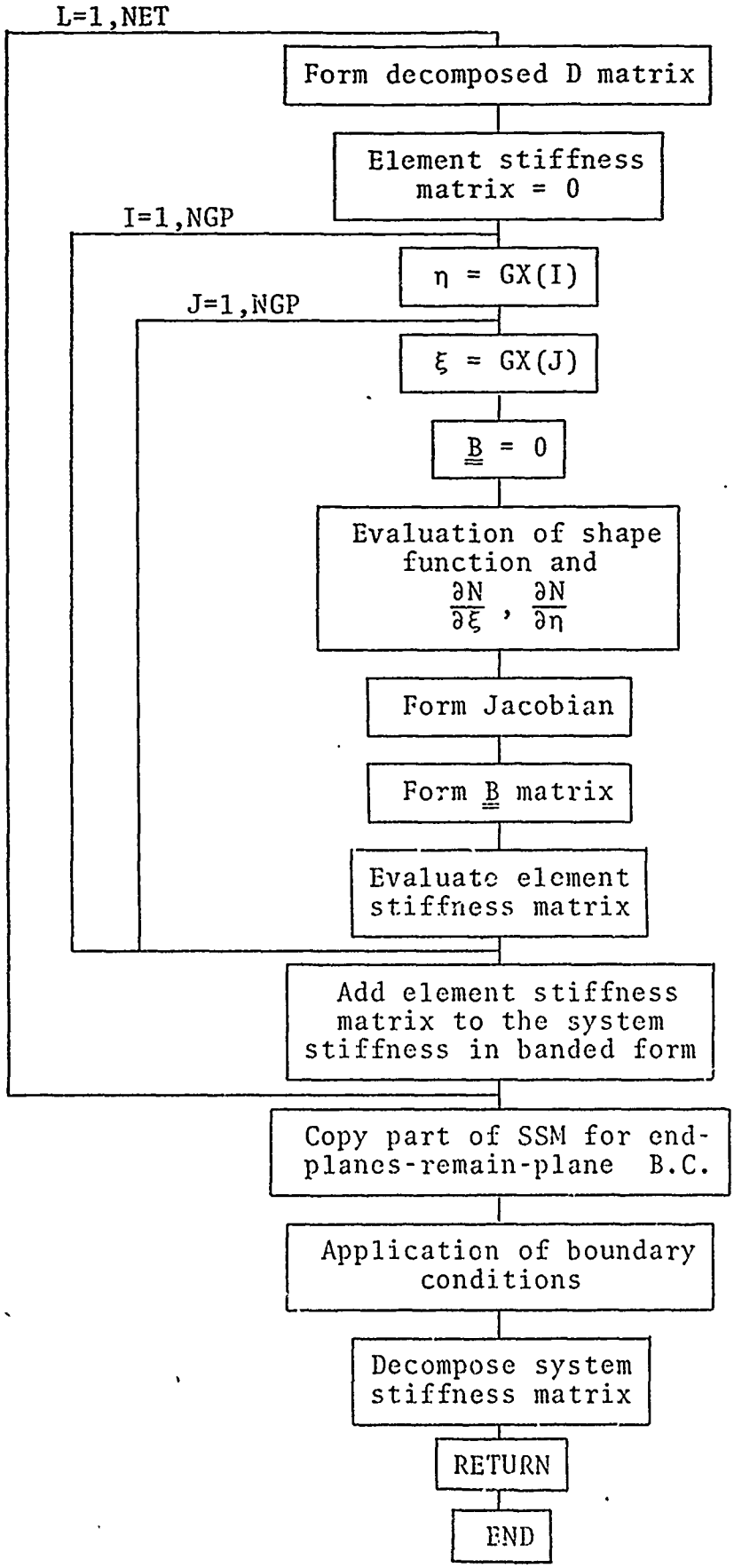


Figure 24. Functional Flow Chart on STIFF.

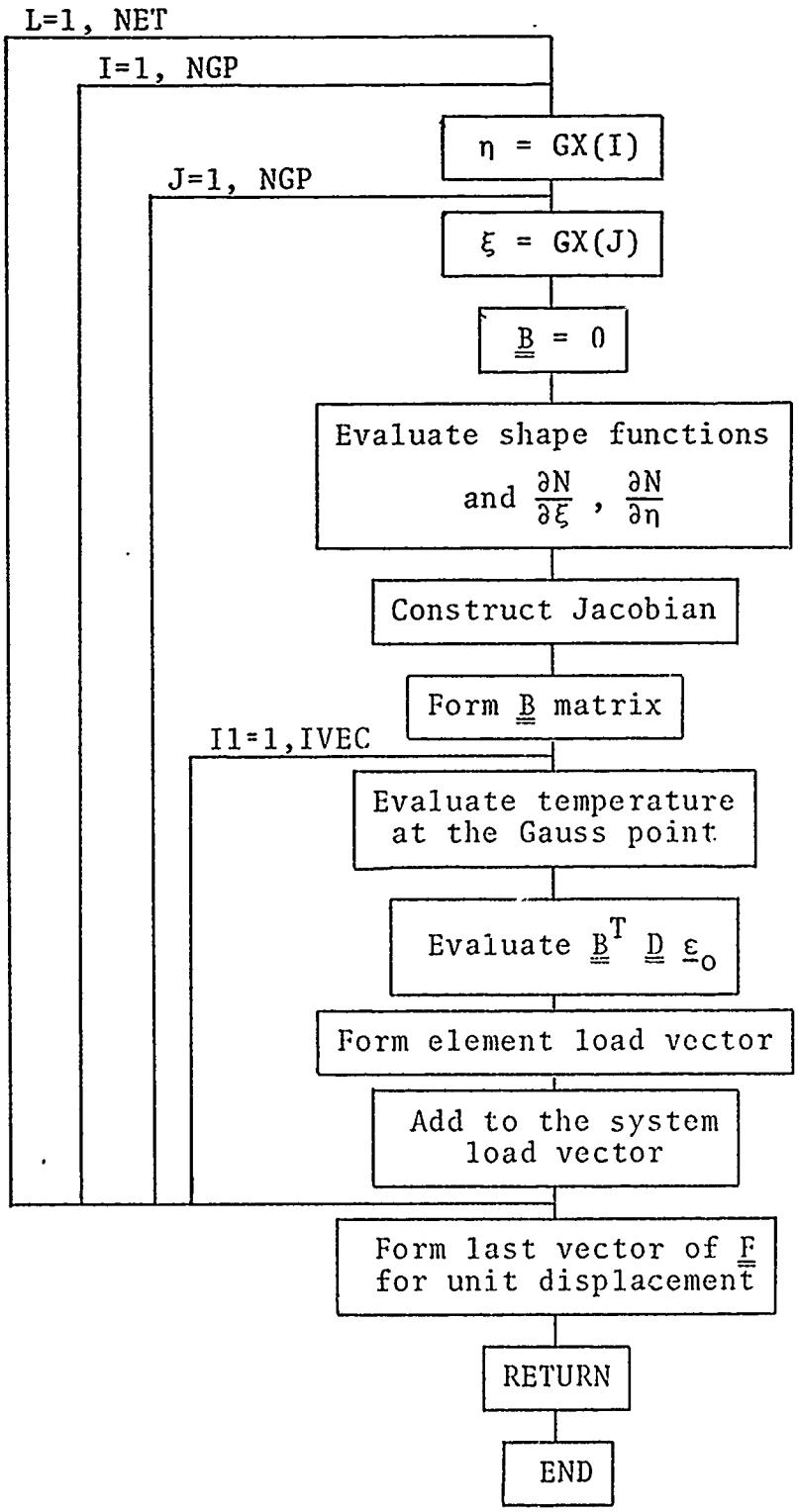


Figure 25. Functional Flow Chart of FØRMF.

11. SUBRØUTINE PRESS

If there is any pressure acting on the system, the pressure load vector is calculated in this subroutine and this vector is placed in the column next to centrifugal load vector, if any, otherwise it will fill the position designated for centrifugal load vector in $\underline{\underline{F}}$.

12. SUBRØUTINE DISPL

The displacement vectors are found in this subroutine. Since the stiffness matrix is already decomposed in banded form, then the displacement vectors one after another are obtained by back and forward substitution. The principle of superposition is applied here and finally the axial force, if any, is corrected for the plane-end boundary condition.

13. SUBRØUTINE STRESS

For every problem this subroutine is called by the main program once to evaluate the stresses at the points $(\eta=\pm 1,\ \xi=\pm\frac{1}{\sqrt{3}})$ of each element. The transient stresses are calculated and printed for each displacement vector. For each problem the maximum mean stress and the maximum octahedral shearing stress, together with the corresponding times and locations, are printed.

14. LIMITATIONS

In the process of the development of the program discussed so far, it has been intended that all of the calculations and storage of data occur in the computer without

using any external devices such as disks or magnetic tapes so as to be able to solve any sizable problem in relatively short time.

For this reason the following limitations (Table VI) are set forth which give an overall size of 450K bytes to the program.

TABLE VI

MAXIMUM VALUES FOR PROGRAM PARAMETERS

NBAND	Half band-width of the system stiffness matrix	66
NMAT	Total number of different materials	5
NET	Total number of elements	40
NNT	Total number of nodes	149
NCFØT	Total number of nodes in contact with outside fluid	37
NCFIT	Total number of nodes in contact with inside fluid	37
NNRE	Total number of nodes of right-hand end	37
NNLT	Total number of nodes constrained against any longitudinal motion	37
NNRT	Total number of nodes constrained against any radial motion	37
IVEC	Total number of temperature vectors stored for thermal stresses	20
NRAMPØ	Total number of ramps for outside flow	15
NRAMPI	Total number of ramps for inside flow	15
NPNT	Total number of pressure nodes	37

LIST OF REFERENCES

1. Zienkiewicz, O. C., The Finite Element Method in Engineering Science. McGraw-Hill, 1971.

2. Arpaci, Vedat S., Conduction Heat Transfer, Addison-Wesley Publishing Company, 1966.

3. Collatz, Lothar, Numerical Treatment of Differential Equations. Third edition. Springer-Verlag, New York Inc., 1966.

4. Private communication from Bruce M. Irons to Robert E. Newton.

5. Chapman, J., Heat Transfer, Second edition. MacMillan Company, 1967.

6. Timoshenko, S. and Goodier, J. N., Theory of Elasticity, Second Edition, McGraw-Hill, 1951.

7. Fung, Y. C. Foundations of Solid Mechanics. Prentice-Hall, 1965.

8. Irons, B. M., International Journal for Numerical Methods in Engineering, Vol. 2, No. 1, p. 5-35, January-March 1970.

9. Boley, Bruno A. and Weiner, Jerome H., Theory of Thermal Stresses. John Wiley and Sons, Inc., 1962.